PROGRAMMABLE STOCHASTIC ASSEMBLY OF MICROSCALE COMPONENTS

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Michael Thomas Tolley

January 2011

PROGRAMMABLE STOCHASTIC ASSEMBLY OF MINIMALISTIC COMPONENTS

Michael Thomas Tolley, Ph. D.

Cornell University 2011

Stochastic fluidic assembly is an approach to small scale fabrication that serves as an alternative to both top-down pick-and-place assembly and bottom-up self-assembly. It avoids the complications of top-down assembly by relying on stochasticity in the environment for component transportation and local self-assembly forces for component positioning. However, unlike pure self-assembly approaches, stochastic assembly is dynamically programmable and can assemble arbitrarily specified (nonregular, nonrandom) structures.

The work presented here advances the state of the art of stochastic fluidic assembly with contributions in four areas. The first area of contribution is the demonstration of stochastic fluidic assembly at the microscale. Previous work in dynamically programmable stochastic assembly (fluidic or otherwise), has been done at the cm or dm scales. The work in Part I of this dissertation describes experiments that demonstrate the assembly of arbitrary structures composed of up to 10 microcomponents, the first steps to adding functionality to the components, and hierarchical approaches for the acceleration of assembly. These advances were achieved by taking an approach that minimizes the complexity of the components required for assembly.

The second area of contribution, presented in Part II, is the demonstration of robust 3-D assembly at the cm scale. Previous stochastic assembly approaches have

demonstrated either 2-D assembly, or 3-D planar assembly at the dm scale. Again, these contributions were made possible by developing a stochastic assembly that minimizes the required module functionality.

Part III of this dissertation presents work relating to the third area of contribution, computationally efficient simulation of stochastic fluidic assembly. The ability to simulate stochastic fluidic assembly is invaluable in system design, assembly algorithm development, and assembly time and error prediction. However, the standard method of simulating fluid-structure interaction—involving Computational Fluid Dynamics (CFD)—is very computationally expensive. In Part III, a custom simulator is presented that makes simplifications where possible, and is tested against CFD results and experiments.

The final area of contribution, presented in Part IV of this dissertation, is in the development of assembly strategies for stochastic fluidic assembly. This work presents a set of strategies developed and evaluated in simulation. Additionally, a novel approach is presented for stochastic assembly (fluidic or otherwise) that analyzes a target structure and finds valid assembly sequences on-the-fly.

BIOGRAPHICAL SKETCH

Michael Thomas Tolley was born in Chatham, Ontario on May 22, 1981 to Anne and Robert Eric Tolley, and has a brother and a sister, eight and five years older, respectively. Three months after his birth, his father's career as an electrical engineer brought the family to Kingston, Ontario where he was raised. Michael's father's profession, as well as his mother's career as an elementary school teacher, no doubt helped pique Michael's interest in the sciences and engineering. From a young age he dreamt of being "an inventor". He spent a large part of his childhood playing either with construction toys such as Lego and Construx, or with the family's Commodore 64C.

Michael attended high school at the Kingston Collegiate and Vocational Institute (KCVI). In addition to his excellent science, math, and computer classes at KCVI, Michael was fortunate enough to be able to attend extracurricular programs during high school such as the Shad Valley summer program at McMaster University that helped develop his interest in engineering and academics.

Michael graduated from KCVI in 2000, and was accepted to the Honours Mechanical Engineering program at McGill University, Montreal, Quebec. As in high school, it was not only the Mechanical Engineering program at McGill that fostered his excitement for engineering research, but also his experiences with the solar car team and with a four-person engineering team that competed at the school, provincial and national levels. Inspired by these experiences, Michael joined the Ambulatory Robotics Lab at McGill's Center for Intelligent Machines, to conduct research made possible by a Natural Sciences and Engineering Research Council Undergraduate Student Research Award.

Encouraged by his undergraduate advisor, Professor Laurent Mydlarski, and inspired by Professor Hod Lipson's work in robotics, Michael applied to the Mechanical Engineering M.S./Ph.D. program at Cornell University. Professor Lipson's open minded attitude towards foreign students, a TA position in Applied Systems Engineering thanks to Professor Mason Peck, and further help from former Cornell Professor Raffaelo D'Andrea, all enabled Michael to enter the Ph.D. program at Cornell, and begin the work presented here.

To my parents, for encouraging and supporting my academic pursuits,

and especially to my mother for always being there for me.

ACKNOWLEDGMENTS

This work was funded by grants from the U.S. National Science Foundation program for Materials Processing and Manufacturing, the Defense Advanced Research Projects Agency Defense Sciences Office Programmable Matter Program, and the Natural Sciences and Engineering Research Council of Canada, Postgraduate Scholarship program.

I would also like to extend my deepest gratitude to the following people for their support throughout my time at Cornell:

To Hod Lipson for being an inspirational and understanding advisor, for celebrating my successes and for encouraging me through my frustrations.

To David Erickson for advising me throughout my microscale work, and for encouraging me to be bold.

To my collaborators, Mekala Krishnan, Michael Kalontarov, Jonas Neubert, and Jon Hiller, for participating in my research and sharing their creativity.

To the current and past members of the Cornell Computational Synthesis Lab who have given their help and advice throughout my work. In particular, I would like to thank Michael Schmidt, Dan Cohen, Victor Zykov, Evan Malone, Aaron Lenfestey, Stephane Constantin, Denise Wong, Juan Zagal, Ricardo Garcia, John Rieffel, John Amend, Rob McCurdy, Jeffrey Lipson, and Daniel Ly for being great colleagues and great friends.

To the members of the Erickson Lab who have given their help and advice with my microscale work. In particular I would like to thank Bernardo Cordovez and Sudeep Mandal for being there from the start of my work at Cornell.

To all my colleagues who have offered their technical advice and their friendship, especially Jay Schuren, Justin Atchison, Edgar Cuji, Vishal Tandon, Brett Streetman, Michele Carpenter, Shan Mohuiddin, and Michael Sherback.

To the faculty of Cornell University, especially Professor Doug James for bringing a different perspective to my committee and to Professors Mason Peck and Raffaello D'Andrea, for making it possible for me to come to Cornell.

To the staff of Cornell University, especially Marcia Sawyer who went over and above the call of duty in helping me navigate my Ph.D.

To the helpful staff at the Cornell NanoScale Science & Technology Facility (CNF), and to Andrew Baisch, for being a great REU student.

To my good friends Chris McQuinn, Scott Chisholm, and Tracy Chew for keeping me grounded and sane during my time away from Canada.

To Diana Chang, for her friendship and support, and for her tireless proofreading.

To my family, for their encouragement and unconditional love.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

CAD – Computer-Aided Design

CFD – Computational Fluid Dynamics

CNF – Cornell NanoScale Science & Technology Facility

CPU – Central Processing Unit

DRIE – Deep Reactive Ion Etch

FSM – Finite State Machine

GUI – Graphical User Interface

MEMS – Microelectromechanical System

μTAS – Micro total analysis systems

NEMS – Nanoelectromechanical System

ODE – Open Dynamics Engine

PDMS – Polydimethylsiloxane

PWM – Pulse Width Modulation

SEM – Scanning Electron Microscope

SOI – Silicon-On-Insulator

USB – Universal Serial Bus

INTRODUCTION

Imagination will often carry us to worlds that never were.
But without it we go nowhere.

- Carl Sagan (1980)

Motivated by visionaries like Carl Sagan, this dissertation aims to take an idea from science fiction and bring it one step closer to reality. The idea is a set of fundamental building blocks that can be made to assemble into any desired shape, to serve any required task. Once the task is complete, the assembled object can then be broken back down into its constituent building blocks. Essentially, this idea amounts to a form of programmable matter (McCarthy, 2000), that can be made to take on any shape and properties necessary to fill a particular function.

Of course, one might argue that we already have an amazing range of engineering processes for manufacturing an even larger range of products, so why invest effort in programmable matter? This approach has at least four potential advantages over traditional manufacturing methods, the first of which is adaptability. Programmable matter would be very useful in situations where you can't predict ahead of time what tools you will need to accomplish a task. For example, consider the problem of packing for a mission to Mars. It may not be possible to predict everything you'll need when you arrive, and it may not be practical to wait for supply shipments once you've found out. The ideal solution would be to bring along vats of building blocks that can be assembled into anything you need when you arrive.

A second potential advantage of this approach to manufacturing is reduced waste. The current approach is to discard complex objects such as electronics when they are damaged or no longer needed. If an object made of programmable matter becomes

damaged, the damaged building blocks can be removed and replaced, as necessary. If the object is no longer needed, the entire thing can be disassembled into the basic building blocks for later reuse.

The third potential advantage of programmable matter is a reduction in the cost of creating new or customized devices. Under the current paradigm, there is a huge upfront expense to setting up a process for manufacturing each new product, which must then be mass-produced in order to realize the economies of scale necessary to make it affordable. Since the alternative proposed here assembles any desired object from a small set of building blocks, the building blocks themselves can be mass-produced inexpensively without constraining the final product.

The fourth advantage is probably the most important reason to pursue any such transformative technology: the unanticipated applications. While it may be apocryphal, Thomas J. Watson is famously quoted as saying in 1943, "I think there is a world market for maybe five computers" (Maney, 2003). True or not, this quote highlights the fact that it is often difficult to anticipate the ways in which such an open-ended technology may be useful.

With these potential benefits in mind, this dissertation takes the first steps toward realizing programmable matter by examining the problem of assembly. In order to assemble any object from discrete building blocks, it is necessary that they be small enough to achieve a high resolution in the final product. To this end, our goal is to assembly building blocks of 100μm to 500 μm (corresponding to a 2-D printing resolution of approximately 50 to 250 dpi). However, traditional top-down assembly approaches break down at this scale as forces like friction and surface tension become dominant with the increase in the ratio of surface area to mass (Böhringer, Fearing, & Goldberg, 1999). As a result, the infrastructure required for top-down assembly becomes increasingly large and complex as the size of the components decreases.

Taking inspiration from nature, many researchers have taken the opposite approach: bottom-up self-assembly (Philip, 1996; Whitesides & Grzybowski, 2002; Glotzer, 2004). However, in order to assemble structures with arbitrarily specified (nonregular, nonrandom) geometry, these approaches require selective affinities between each pair of components, analogous to the interlocking shapes of pieces in a jigsaw puzzle (Bowden, Terfort, Carbeck, & Whitesides, 1997; Srinivasan, Liepmann, & Howe, 2001; Winfree, Liu, Wenzler, & Seeman, 1998; Rothemund, 2006). Like a jigsaw puzzle, these unique affinities only assemble in one way and the pieces must all be redesigned to achieve a new target shape. Thus, a dynamically programmable self-assembly approach capable of fabricating arbitrary structures from microscale building blocks has yet to be discovered.

Inspired by biology, we take an approach halfway between the extremes of pure top-down pick-an-place assembly and pure bottom-up self-assembly. This approach is called stochastic assembly since it relies on stochastic environmental motions for the transportation of building blocks. This approach is inspired by nature where—at small scales—products will often be transported by diffusive processes based on Brownian motion. However, these random motions are directed in order to achieve the desired assembly. This stochastic assembly approach has been demonstrated in modular robotic systems at the centimetre and decimetre scales (White, Zykov, Bongard, & Lipson, 2005; Griffith, Goldwater, & Jacobson, 2005; Napp, Burden, & Klavins, 2006; Gilpin, Kotay, Rus, & Vasilescu, 2008). The motivation for much of this previous work was to develop assembly approaches that would scale down in size. However, in addition to being at least two orders of magnitude larger than our target size of 500 μm, all of this previous work was performed in two dimensions, with the exception of Gilpin et al., who focussed on an alternative approach based on disassembly.

Fortunately, there is one approach with the potential to solve both the problems of scaling the building blocks down two orders of magnitude and of moving the assembly out of the plane and into the third dimension: further simplification of the building blocks. This is the approach taken in the work presented in this dissertation. Of course, further component simplification leads to a more demands on the assembly environment, and this is one of the fundamental tradeoffs examined in this work.

This dissertation is split into four parts, each of which contains chapters relating to a different aspect of the programmable stochastic assembly of microscale components or components simple enough to be viable at the microscale. Part I is composed of experimental work relating to microscale assembly. Within it, Chapter 1 describes experiments in the dynamically programmable fluidic assembly of microscale tiles in two dimensions. This work is based primarily on a paper published in Applied Physics Letters (Tolley, Krishnan, Erickson, & Lipson, 2008), as well as a conference paper presented at the μTAS conference in 2006 in Tokyo (Tolley, Zykov, Erickson, & Lipson, 2006). Chapter 2 expands on this work by analyzing the mechanical and electrical interfacing between components required to improve their assembly and potentially add the ability to program their functionality. Chapter 2 is based primarily on a paper presented at the 2008 IEEE International Conference on Microelectromechanical Systems (Tolley, Baisch, Krishnan, Erickson, & Lipson, 2008). Chapter 3 is composed of unpublished work on the topic of hierarchical microfluidic assembly as a potential route to fab-on-a-chip.

Part II of this dissertation contains one chapter (Chapter 4) that focuses on experimental three-dimensional stochastic fluidic assembly. In order to deal with the challenges of three-dimensional assembly, the experiments described in this chapter are conducted with 1.5 cm scale components. However, these components are kept as simple as possible in order to be viable at the microscale. Chapter 4 describes the

design and testing of the experimental system for 3-D stochastic fluidic assembly, as well experiments demonstrating manipulation, planar assembly, repair, robust 3-D assembly, and parallel hierarchical assembly. The material in this chapter is based on a paper published at the 2010 International Conference on Robotics and Automation (Tolley & Lipson, 2010), as well as unpublished work.

Part III of this dissertation contains a single chapter (Chapter 5) that relates to the simulation of stochastic fluidic assembly. This chapter presents a custom computationally efficient environment for simulating the stochastic fluidic 3-D assembly of components with limited complexity. Basic predictions of the components interactions and assembly rates are compared against precise but expensive CFD simulations as well as experiments, and important system parameters are identified. This work is based on a paper published in the IEEE Transactions on Robotics (Tolley, Kalontarov, Neubert, Erickson, & Lipson, 2010).

Finally, Part IV of this dissertation contains two chapters that investigate the problem of controlling stochastic fluidic assembly. The first, Chapter 6, uses the simulation environment of Chapter 5 to analyze the effectiveness of a set of static control strategies, and is also based primarily on a paper published in the IEEE Transactions on Robotics (Tolley, Kalontarov, Neubert, Erickson, & Lipson, 2010). Chapter 7 presents work that takes an on-line approach to planning for stochastic assembly. Instead of trying to predict where and how problems can arise that would prevent perfect assembly, this work analyzes the remaining structure to be assembled and continually samples potential assembly sequences to select where next to add modules. Chapter 7 is based on a manuscript currently under review by the International Journal of Robotics Research.

PART I: MICROSCALE EXPERIMENTS

CHAPTER 1: DYNAMICALLY PROGRAMMABLE FLUIDIC MICROASSEMBLY

Abstract

A major challenge in fluidic assembly is the dynamically programmable fabrication of arbitrary geometries from basic components. Current approaches require predetermination of either the assembly machinery or the component interfaces for the specific target geometries. In this chapter we present an alternative concept that exploits self-assembly forces locally, but directs these forces globally, allowing fabrication and manipulation of target structures without tailoring the substrate or interfaces. By controlling the flow in a microfluidic chamber, components are directed to their target locations where local interactions align and bond them. Following this approach, we demonstrate and quantify the experimental assembly of structures composed of two to ten components.

Introduction

There has been great interest recently in self-assembly (Philip, 1996; Whitesides & Grzybowski, 2002; Glotzer, 2004) as an avenue for the fabrication of functional microscale (Jacobs, Tao, Schwartz, Gracias, & Whitesides, 2002) and nanoscale (Vlasov, Bo, Sturm, & Norris, 2001; Parviz, Ryan, & Whitesides, 2003) devices. However, in order to assemble structures with arbitrarily specified (nonregular, nonrandom) geometry, bottom-up assembly approaches generally require selective intercomponent affinities for each pair of components using mechanisms such as free surface energy minimization with shape recognition (Jacobs, Tao, Schwartz, Gracias,

& Whitesides, 2002; Bowden, Terfort, Carbeck, & Whitesides, 1997; Srinivasan, Liepmann, & Howe, 2001), electrostatic interactions (Grzybowski, Winkleman, Wiles, Brumer, & Whitesides, 2003), and DNA base pairing (Winfree, Liu, Wenzler, & Seeman, 1998; Rothemund, 2006). These static affinities are analogous to the interlocking shapes that uniquely determine piece positions in a jigsaw puzzle. This "jigsaw puzzle" approach, however, has several challenges: First, the components must be redesigned specifically for each new target structure. Second, as the target structure's complexity increases, the number of distinct affinity patterns required also increases. This causes the selectivity of intercomponent affinities to decrease due to a fixed affinity pattern space. The result is an increased probability of assembly errors and a decrease in assembly rates due to the relative dilution of each component type. An alternative approach (Chung, Park, Shin, Lee, & Kwon, 2008) assembles a target structure using a set of rails to guide components along predetermined paths. This approach shows promise in its ability to accommodate a large variety of target geometries; however any given rail architecture places constraints on the set of attainable target structures and manipulation paths. For example, railed substructures cannot easily be rotated to form larger structures or to be used in different configurations. Thus, the rail system itself must be tailored to the target assembly task.

Here we study an unguided assembly process that uses fluidic forces to assemble arbitrary structures from regular components, while avoiding the limitations of a purely stochastic or fixed-rail system. This increased flexibility comes at the expense of more complicated control requirements. Instead of propelling components along guided paths, our system creates the fluid flow conditions that attract components to where they are needed. At this point the regular geometric design of the components causes them to self-align and bond. Robust assembly procedures are used to overcome the stochasticity inherent in such an indirect assembly method. Thus, the assembly is

directed and does not rely on stochastic agitation for component transportation. These features help to overcome the slow assembly rates and assembly errors that inhibit the scalability of current self-assembly processes to the fabrication of arbitrary structures while maintaining the capability of defining the target structure dynamically (i.e. without requiring component or system redesign). Thus our approach combines the opportunities–and challenges–of both the "pure self-assembly" and path-directed approaches.

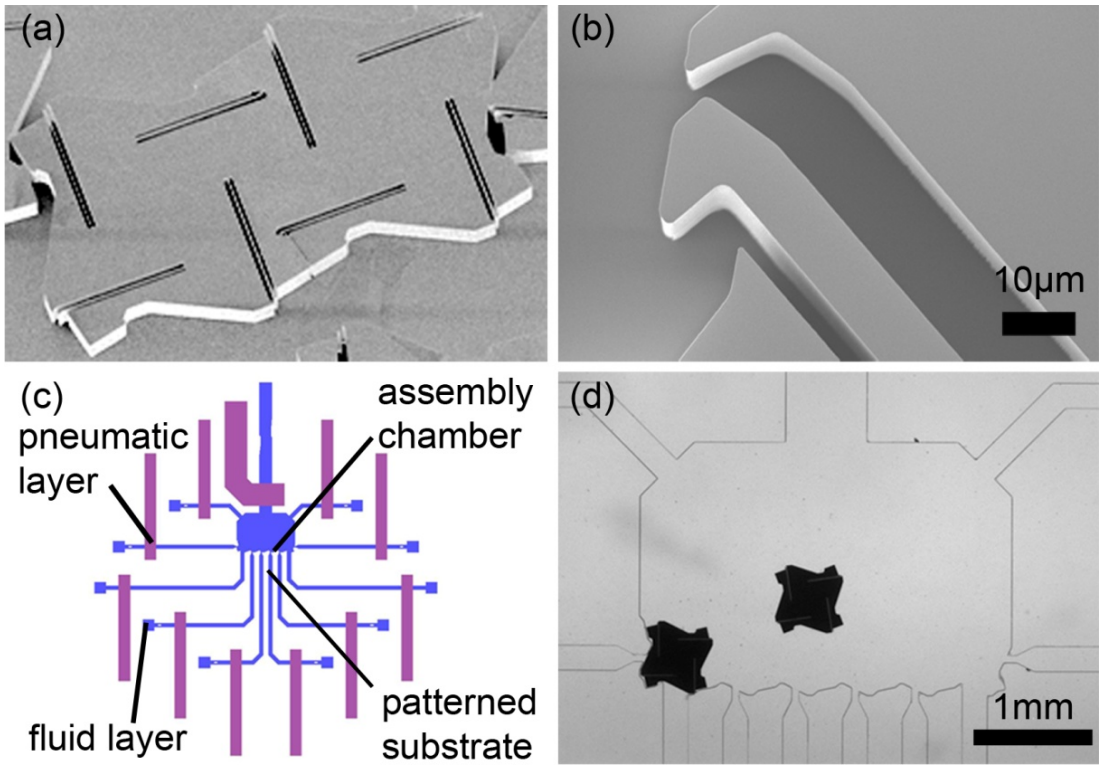


Figure 1.1. Experimental system. (a) Scanning electron microscope (SEM) image of silicon tiles with patterned sides for alignment, held together by compliant latches. (b) Detail of latch design. (c) Computer aided design drawing of multilayer microfluidic assembly chamber with pneumatic valving and intersecting channels for injection and removal of fluid and tiles. (d) Optical micrograph of assembly chamber and two microtiles.

Materials and Methods

We demonstrate our semidirected fluidic assembly process experimentally in two dimensions. Our components are 500 × 500 μm^2 silicon tiles with 30 μm thickness and a single passive latch on each side [Figure 1.1(a)]. Two main features of the tile design facilitate assembly: a regular pattern on the tiles' sides causes adjacent tiles to self-align, while passive latches on each tile side [Figure 1.1(b)] bond tiles together. We achieve indirect control of the positions of our micro-scale components by immersing them in an assembly fluid within a polydimethylsiloxane (PDMS) microfluidic chamber [Figure 1.1(c)–(d)] and directing the fluid flow through the chamber (Figure 1.2). We control the flow conditions by regulating the pressures at eight active openings where microfluidic channels join the assembly chamber – four along the bottom, and two on each side. (As the chamber is mounted horizontally, *bottom* is an arbitrary designation.) A geometric pattern along the bottom of the chamber causes the microtiles to seat in specific locations adjacent to each of the openings, thus forming a "patterned assembly substrate." The remaining three larger openings are used for the introduction of tiles into and extraction of debris out of the chamber.

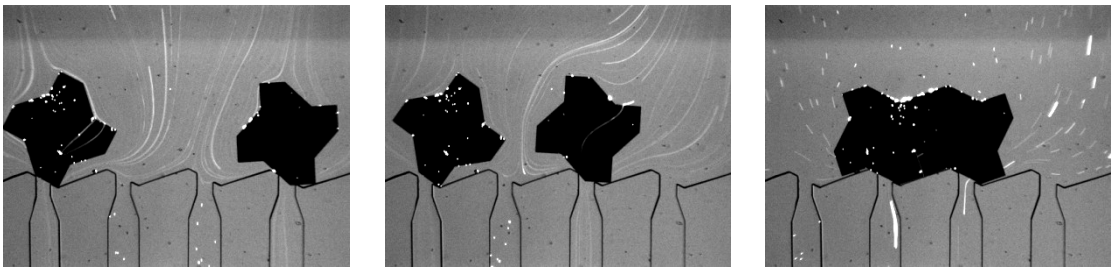


Figure 1.2. Assembly by microfluidic manipulation. Three frames from a video micrograph of an assembly experiment with fluorescent particles added to the flow for visualization. Fluorescent lines indicate direction of fluid flow, while dots represent stagnated flow caused by closed outlet valves.

On- and off-chip valving, achieved using soft lithography (Unger, Chou, Thorsen, Scherer, & Quake, 2000) allow digital control of the pressure (high, low, or closed) at each active opening. This is essentially a form of *boundary condition phenomenological superposition distributed manipulation* as defined by Varsos and Luntz (Varsos & Luntz, 2006). The resulting fluid flow applies hydrodynamic forces and torques to a microtile within the chamber, causing it to be attracted to an opening location on the assembly substrate. Subsequently, the opening and closing of other valves attract other microtiles and bind the components together to form subassemblies. These subassemblies can then be rejected from their seated position and reoriented in the same way as individual components. Thus, with appropriate sequencing of valve openings, a set of subassemblies can be made to form deterministically into an arbitrary structure.

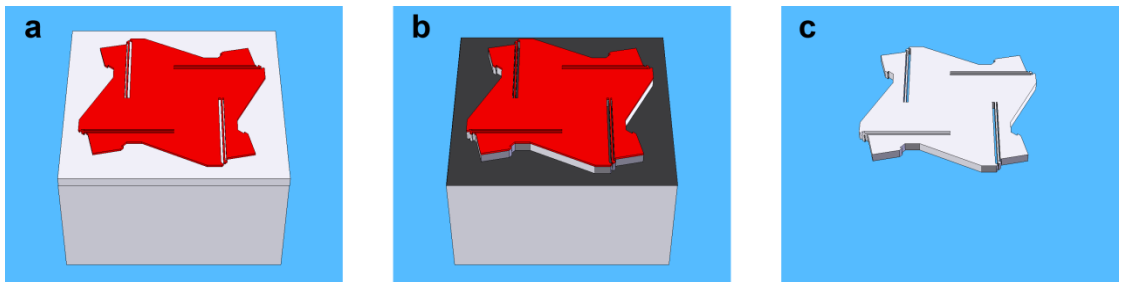


Figure 1.3. Microtile fabrication steps. (a) Patterning of photoresist on silicon-on-insulator (SOI) wafer. (b) Bosch process deep reactive ion etch (DRIE) through device layer. (c) Resist strip and insulator layer wet etch in 49% HF to release the tiles.

Microtile Fabrication

Silicon tiles were fabricated from a silicon-on-insulator wafer with 30 μm device layer and 1 μm buried oxide layer. A 7 μm layer of SPR 220 series photoresist was patterned on the wafer by contact alignment. The tile outline was then etched through the device layer with a deep Bosch etch. The tiles were then released using a wet etch

with 49% HF, and collected by filtration (Figure 1.3).

Chamber Fabrication

The PDMS microfluidic chamber with on-chip valving [Figure 1.1(c)–(d)] was fabricated by multilayer soft lithography. Soft lithography moulds were created by spinning SU-8 2050 photoresist on silicon wafers at 4000 rpm to obtain a ~40 μm thickness film which was patterning with contact alignment and hard baked for durability. Sylgard 184 silicone elastomer base and curing agent were using in a ratio of 20:1 for the thin fluidic layer and 5:1 for the thick pneumatic layer. The fluid layer was spun over its mould at 1250 rpm for a thickness of ~60 μm, while the pneumatic layer was poured approximately 1 cm thick. Both were cured at 80°C for one hour, then bonded together (after punching the pneumatic layer ports), and cured for several hours at 80°C. The fluid layer ports were then punched and the entire chamber was cleaned in a plasma oxidize and bonded to a glass slide.

Component Manipulation and Assembly

The silicon tiles, stored in deionized water, were pipetted into the microchamber via a Tygon tube attached to a large port punched through the PDMS. An assembly fluid – consisting of 0.01% Triton X-100 surfactant by mass – was pumped into the fluidic chamber inlets via smaller Tygon tubes. This pumping was achieved by exposing the fluid to pressurized air at 14 to 42 kPa gauge. On-chip valving was achieved by pumping air at approximately 152 kPa gauge to the pneumatic channels over the fluidic outlet channels. Activation of the high and low pressure lines was controlled off-chip by arrays of solenoid valves and their controllers produced by Fluidigm Corporation. A LabView program was then created to control the timing of the inlet flows and the outlet valving. Outlet valving was switched every 3 to 4 seconds while inlet pressures were switched on in pulses 0.5 to 2 seconds in duration (in order to

overcome static friction).

Results

Using the correct sequence of valve states, we assembled and manipulated elementary structures, which form the fundamental building blocks for more complex assemblies. We demonstrated the repeatable, automated assembly of two-tile constructs and both three-tile polymorphs using open-loop valving control. As a first step, we introduced a single tile into the chamber and marched it along the substrate from one inlet/outlet position to the next [Figure 1.4(a)]. This was achieved by opening the valve at the destination position on the substrate to atmospheric pressure and pumping fluid into the chamber from a combination of the remaining openings. Using this automated open-loop control strategy, the tiles moved one position in the desired direction 96% of 356 attempts [Figure 1.5(a)]. During 99% of these motions, the tiles rotated 90° when moving from one outlet to the next. Experiments in the manipulation of assembled tile pairs showed similar trends [Figure 1.5(a)].

The next step towards the assembly of complex shapes was to bring two tiles together and induce them to latch. An automated, open-loop valving sequence was developed to attract two tiles to the substrate, bring them toward one another, and induce them to latch together [Figure 1.4(b)]. The tiles always began in the same initial positions, but their orientations were not tightly controlled. Valving configurations for this 60 s sequence were changed on a 4 s cycle. The tile pair was subsequently moved to verify assembly. This control sequence (strategy 1) resulted in an assembled pair 10 out of 30 consecutive experiments [Figure 1.5(b)]. In the remaining experiments, no assembly occurred.

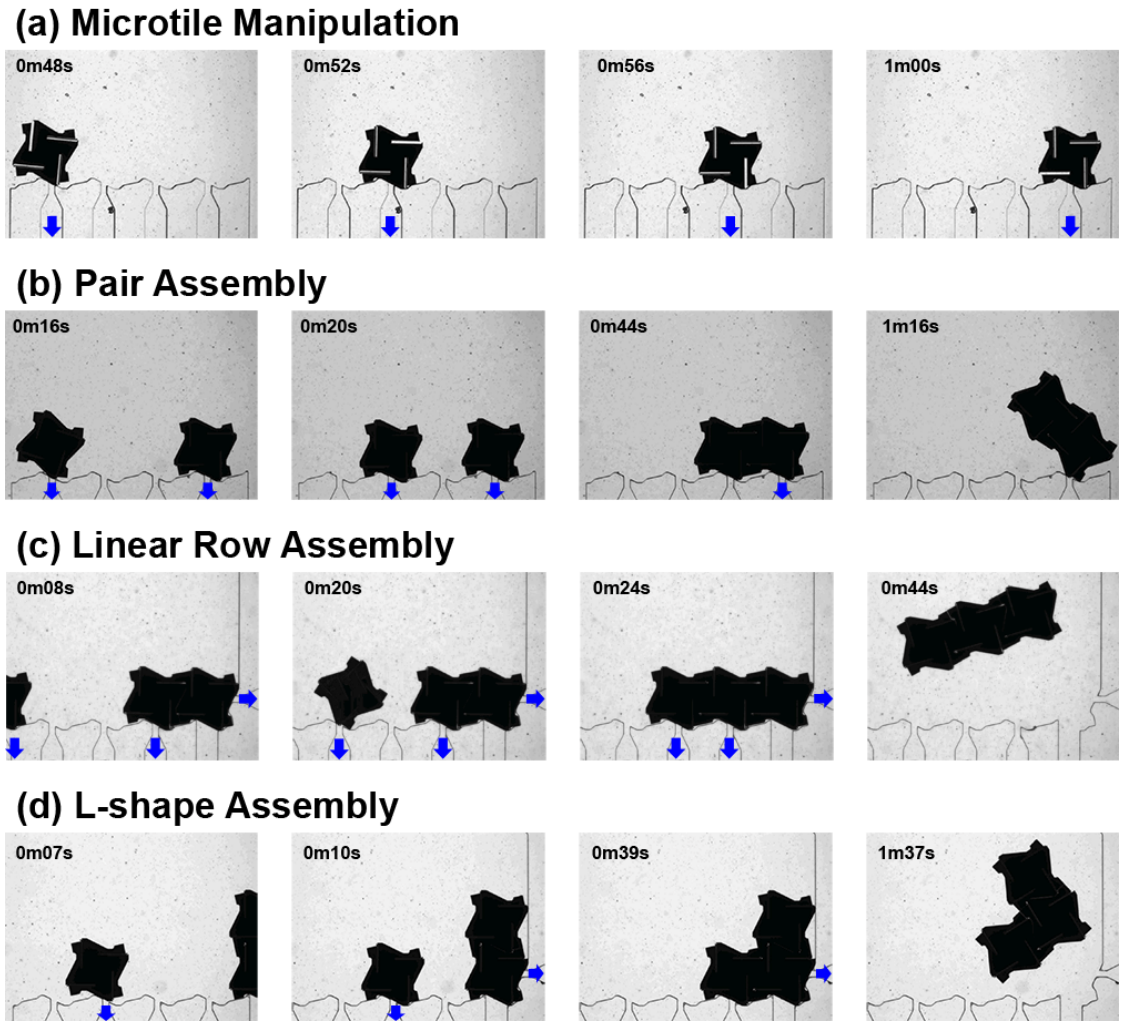


Figure 1.4. Microtile manipulation and assembly. Frames taken from video micrographs of (a) an automated microtile manipulation and (b)-(d) three assembly experiments. Timed valve actuation directs pressurized flow into the microfluidic chamber and out the indicated openings. Fluid flow applies hydrodynamics forces to the microtiles, causing them to move and assemble. Alignment patterns and compliant latches cause adjacent tiles to self-align and bond together. The valving sequences determine the final structures.

As a demonstration of the next layer of assembly, we fabricated three-tile structures in the two possible polymorphs: a linear row and an L-shape. In each case we began with a single tile and a latched pair assembled using valving sequence such as that used in Strategy 1. For linear row assembly (strategy 2), we began with the assembled pair horizontal on the substrate and attracted the third tile next to the

assembled pair [Figure 1.4(c)]. The L-shape assembly sequence (strategy 3) began with a vertical assembled tile pair and attracted a third tile to be attached at the base [Figure 1.4(d)]. In both cases any structures assembled at the end of the sequence were released from the substrate to test for assembly. Strategy 2 assembled a three-tile row 20 of 30 consecutive trials while strategy 3 assembled an L-shape 10 of 30 trials [Figure 1.5(b)]. More importantly, strategy 2 *never* resulted in the assembly of an L-shape and strategy 3 never resulted in the assembly of row.

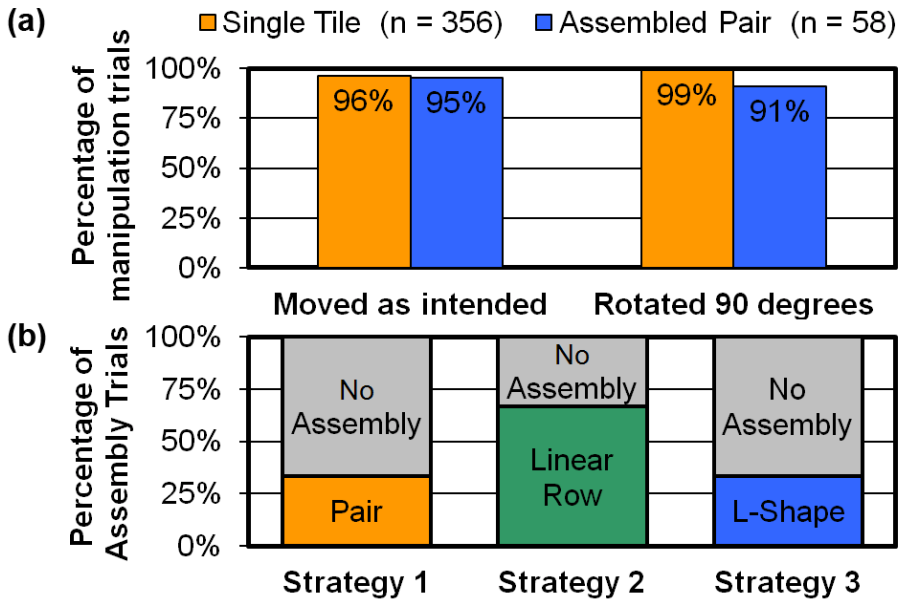


Figure 1.5. Experimental results. (a) Microtile Manipulation: percentage of automated manipulation experiments in which a single tile or assembled tile pair moved and rotated as anticipated. (b) Deterministic assembly: resulting structures at the end of 30 consecutive automated assembly experiments for each of three different open-loop assembly strategies (valving sequences).

Thus, we found our strategies to be *deterministic* in the sense that they never resulted in false assemblies. In all of the cases where strategies 2 and 3 were not successful, the end products were the same as the initial components: a latched pair and a single tile [indicated by *No Assembly* on Figure 1.5(b)]. Of course, the success

rates of any of these three strategies could be increased arbitrarily high by simply repeating the valve sequence (at the cost of longer assembly times).

Discussion

The latches used in these experiments were designed for relatively low assembly and disassembly energies. This allowed the tiles to latch easily but also meant that many assemblies came apart in experiment, reducing the assembly success rates. We also tested stiffer latches which permanently bonded tiles together (Figure 1.6). However, the stiffer latches also made assembly more difficult and complicated the iteration of experiments.

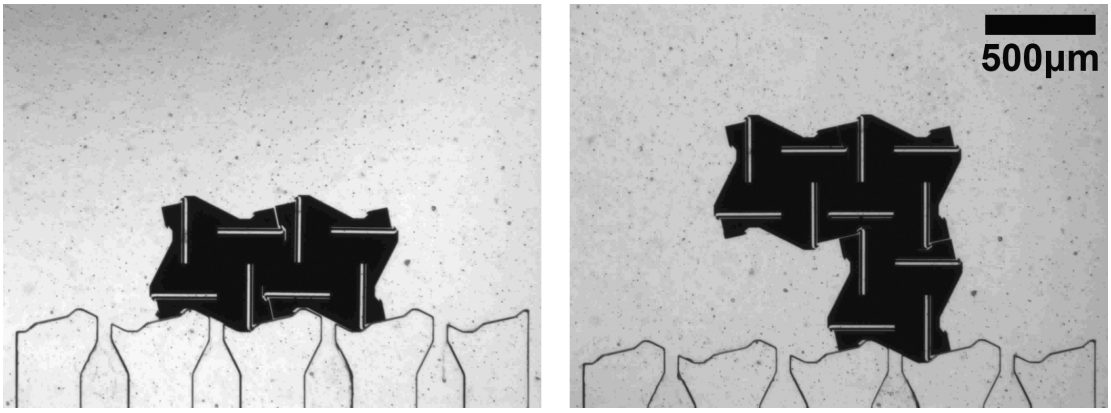


Figure 1.6. Alternative latch design experiments. Frames taken from video micrographs of the assembly of a tile pair and L-shape from stiff-latch microtiles. These structures could not be disassembled with the maximum achievable fluidic forces in the channel.

Alternative valve control strategies are possible to facilitate the assembly of larger structures and to increase assembly rates. While the valving timings and sequences of the open-loop controllers employed here could be further tuned to improve assembly speed and robustness, another approach is also promising: closed-loop feedback control. An automated controller with image processing could be used to identify–and

correct for–assembly errors and improve the system's efficiency and reliability. We demonstrated this closed-loop approach with user-controlled valving guided by visual feedback. In this manner we assembled symmetric structures composed of four to ten components (Figure 1.7), as well as a number of pairs of complementary, mirror-image structures (Figure 1.8).

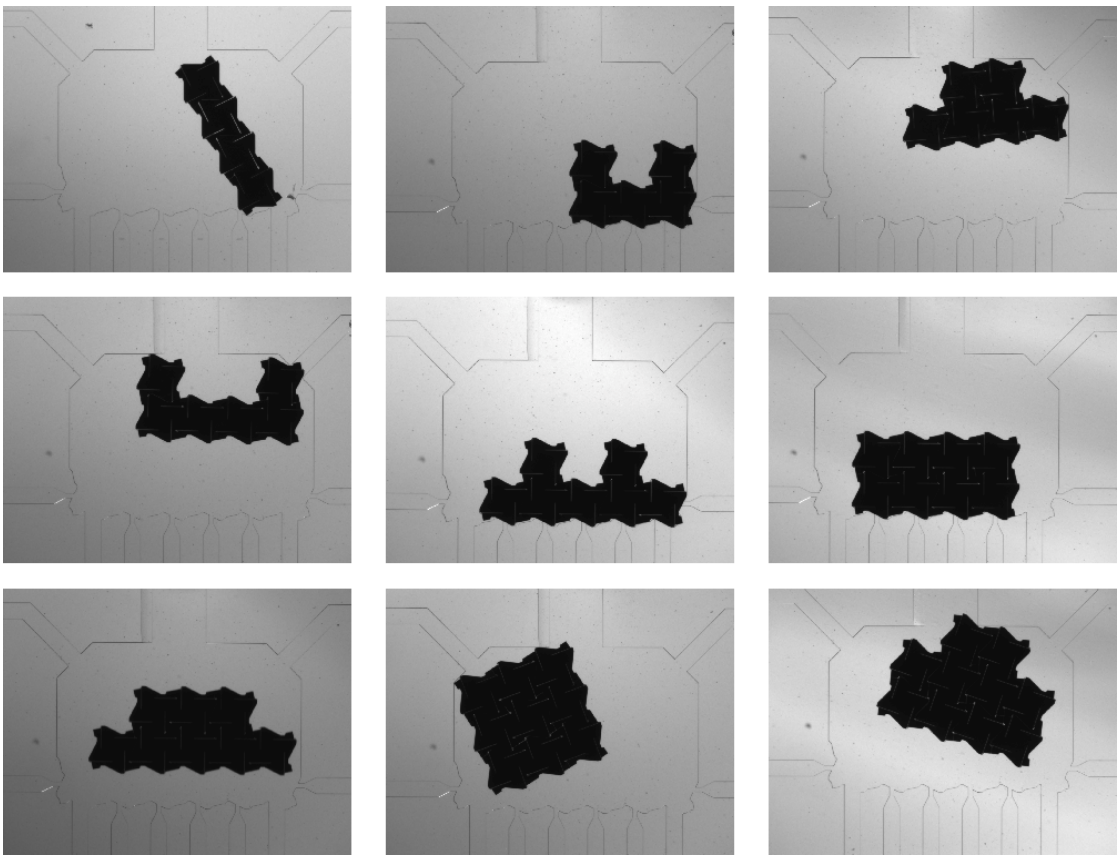


Figure 1.7. Larger Assemblies. Frames from video micrographs of the assembly of symmetrical structures composed of four to ten components. Valve switching was controlled manually with visual feedback.

In the extension of our assembly approach to larger structures, it is important that our system scales well with large numbers of components. One important condition for scalability is the capability of parallel assembly. For large target structure sizes fabricating by adding components sequentially takes a prohibitively long time.

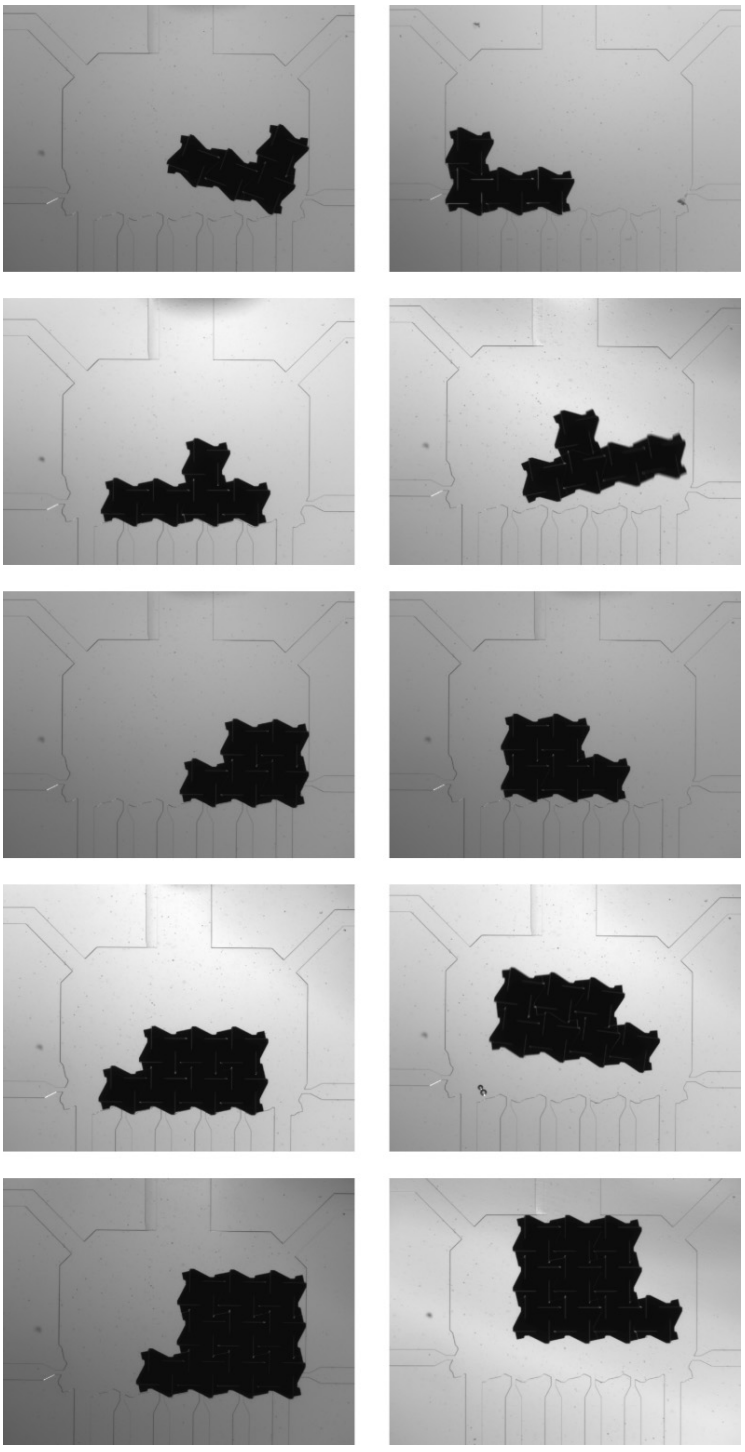


Figure 1.8. Complementary Shapes. Frames taken from video micrographs of the assembly of complementary, mirror-image shapes composed of four to ten components. The assembly of these structures was directed by user-operated valving with visual feedback.

However, such a structure may be feasibly manufactured by dividing the work among multiple assembly sites working in parallel, and then bringing together the produced subassemblies. Our system lends itself well to this kind of parallel assembly. In experiment we have demonstrated that our system manipulates assemblies nearly as effectively as single components [Figure 1.5(a)]. One could imagine a hierarchical approach in which the six-tile-wide substrate of our experimental system was expanded to a network of intercommunicating assembly sites with new components attracted where they are needed. Subassemblies would then be passed on from one site to the next and assembled repeatedly into structures of increasing size until the final structure was completed.

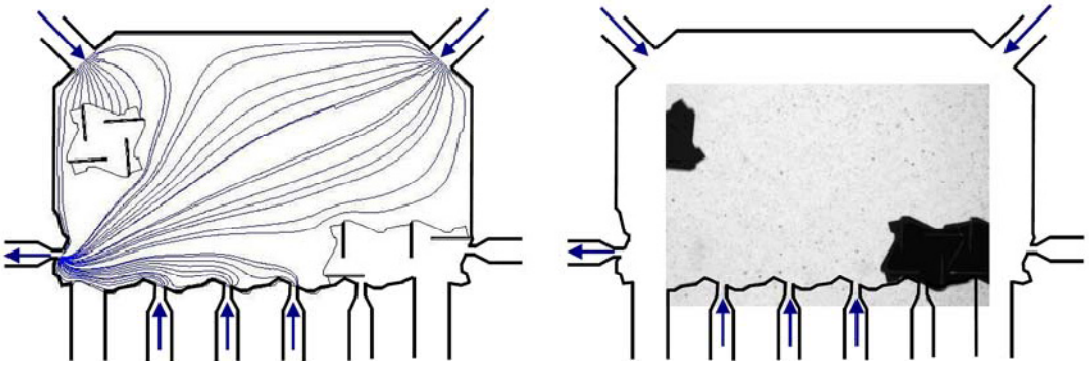


Figure 1.9. Comparison of simulation and experiment. The tile motion in simulation (left) was found to match that observed in experiment (right). The arrows indicate the fluid inlets and outlets in the system.

Analytical Model and Simulations

In collaboration with this work, Mekala Krishnan has developed a simulation of the fluid-tile interactions in the microfluidic assembly systems described in this chapter [(Krishnan, Tolley, Lipson, & Erickson, 2008), Figure 1.9]. The simulation is based on a fluid mechanical analytical model of the system and is used to predict assembly characteristics for various alternative system designs. The experiments

described here were used as a test case for the validation of this simulation. Please refer to (Krishnan, Tolley, Lipson, & Erickson, 2008) for more details on this work.

System Design Choices

As with any open-ended design project, we were faced with many options in designing our fluidic microassembly system. Due to the rapid prototyping nature of microfluidic systems, we were able to investigate many of these options experimentally. In this section we discuss a couple of these design choices and the alternatives that we also investigated.

Tile Design

One of the major trade-offs we considered had to do with the tile design. On the one hand, we wanted any pair of faces on adjacent microtiles to mate (regardless of orientation). For tiles with four sides, this can be achieved using a design with four-fold rotational symmetry (i.e. a design that is identical to itself when rotated 90° in the plane about its centre). The design seen in Figure 1.1 has this property.

A second desirable property is flip-invariance, since it is very difficult to control the orientations of the microtiles as they enter a microfluidic chip. The tiles seen previously do not have this property and will not mate with flipped versions of themselves [see Figure 1.10(a)]. It is possible to design tiles that are flip-invariant using a design that is symmetric about an axis in the plane [see Figure 1.10(b)]. However, this additional requirement is inconsistent with four-fold rotational symmetry and leads to tiles with gendered side patterns. Thus, a given tile side will only mate with half of an adjacent tile's sides. This can significantly complicate assembly, as seen in Figure 1.10(b). Nonetheless, we demonstrated microfluidic assembly/disassembly using tiles with this flip-invariant design [Figure 1.10(c)–(e)].

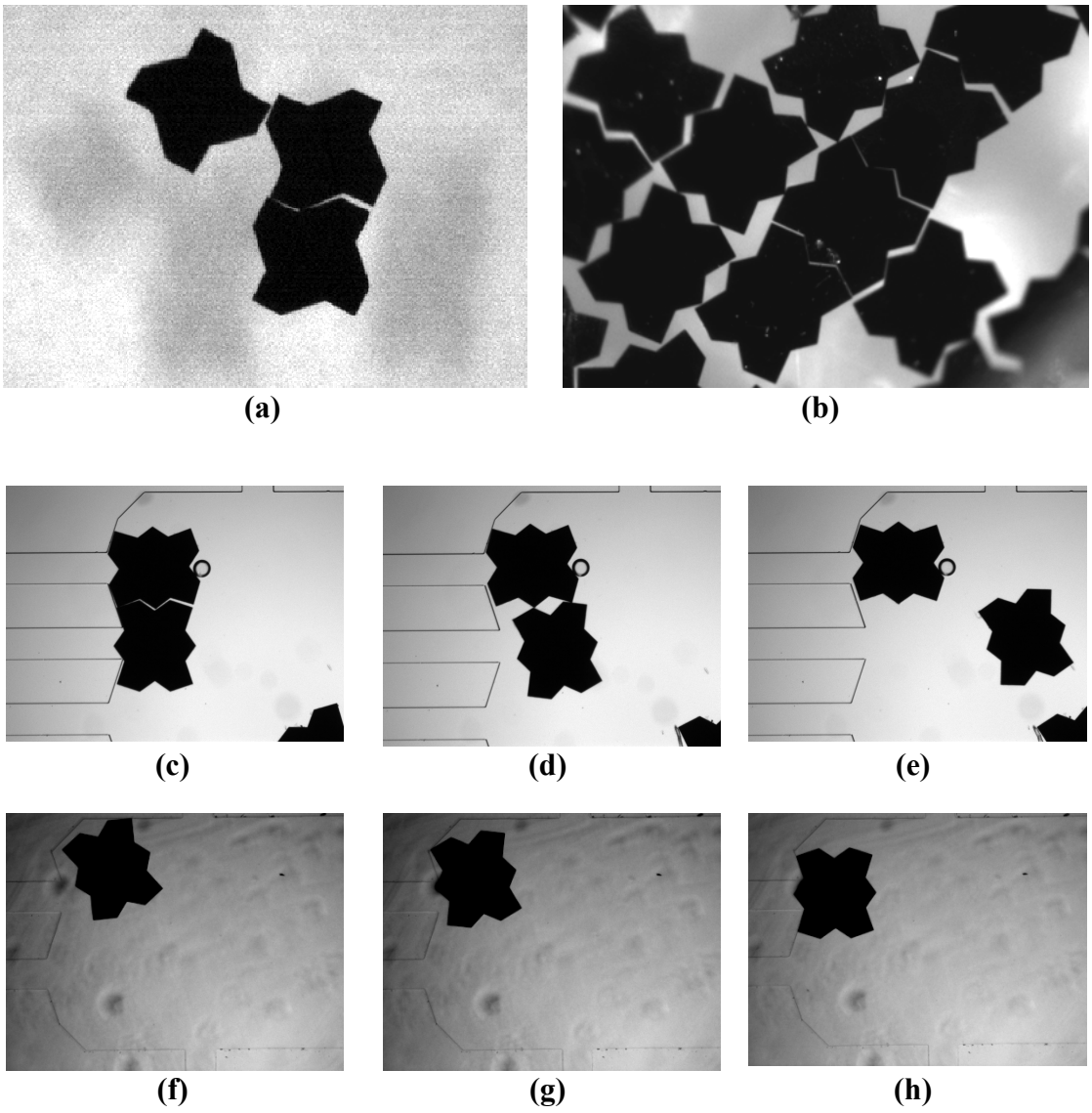


Figure 1.10. System design choices. Frames from video micrographs of assembly experiments. (a) Four-fold rotationally symmetric and (b) flip-invariant tile designs were each found to have advantages and disadvantages for microassembly. Preliminary experiments demonstrated the feasibility of (c)-(e) flip-invariant tiles and (f)-(h) the use of silicone oil as an assembly fluid, although these options were eliminated in favour of more promising alternatives.

Assembly Fluid

Another alternative system design choice we considered was the type of assembly fluid. In the majority of our experiments we used deionized water with a small amount

of surfactant added to reduce adhesion. Another option we considered was the use of silicon oil. This alternative had similar anti-adhesion properties as well as the additional benefit of being non-conductive for increased compatibility with electromechanical systems. We demonstrated the feasibility of silicone oil as an assembly fluid with preliminary experiments [Figure 1.10(f)–(h)]. However, the increased viscosity of oil increased the resistance to tile motion and thus the necessary applied pressures in an undesirable way. Additionally, oil in general was found to be more difficult to work with than water for practical reasons. For example, it reduced the adhesion between the microfluidic chip's PDMS and the hypodermic tubing used to couple to the assembly fluid reservoirs.

Conclusions

The system presented here forms the basis for a dynamically programmable fluidic assembly technique in which regular micro-scale components are assembled hierarchically into arbitrarily complex target geometries. Applications for such a technique include the fabrication of a wide range of micro/nanoelectromechanical (MEMS/NEMS) devices. Additionally, since our assembly mechanism does not rely on special properties of the materials used, a wide range of component materials and assembly fluids is possible (Figure 1.10). For example, silicon microtiles such as those used here could be augmented with sensors, actuators, and electrical interfaces, assembled in a nonconductive fluid such as oil. Alternatively, simple tiles could be fabricated *in situ* by polymerizing photocurable resin (Chung, Park, Shin, Lee, & Kwon, 2008). Combinations of different types of tiles are also possible. This makes our technique particularly appealing for the assembly of devices that require exotic materials with incompatible fabrication processes such as lab-on-chip/bioanalysis systems.

CHAPTER 2: INTERFACING METHODS FOR FLUIDICALLY ASSEMBLED MICROCOMPONENTS

Abstract

In this chapter we present the design and implementation of electrical and mechanical interfaces for fluidically assembled planar MEMS. We discuss the design and fabrication of passive mechanical latches to bond microcomponents together and of conductor layers to establishing electrical connection among assembled components. We evaluate the ability of components with these interfaces to bond together within a microfluidic channel and to establish electrical circuits when assembled. This work supports the development of a novel microassembly approach that bridges the gap between bottom-up self-assembly and top-down direct-manipulation techniques. The ultimate goal of this research is the development of MEMS devices capable of the on-demand self-assembly, repair, and reconfiguration.

Introduction

Self-assembly has shown promise as a practical alternative to pick-and-place assembly at the micro- and nanoscales (Philip, 1996; Whitesides & Grzybowski, 2002). Assembly techniques are necessary for the manufacture of increasingly complex micro/nanodevices with components with incompatible fabrication processes. As scales decrease, however, issues with adhesion, precision, device complexity, and assembly rates reduce the effectiveness of direct-manipulation techniques (Böhringer, Fearing, & Goldberg, 1999). There has thus been a lot of interest in the design of components with interaction forces that cause them to assemble into useful structures.

At the micro-scale, self-assembly has been achieved using primarily surface energy minimization as the diving force with a medium such as solder or adhesive

used to hold the assemblies together (Gracias, Tien, Breen, Hsu, & Whitesides, 2000; Srinivasan, Liepmann, & Howe, 2001). However, the ability of such techniques to assemble arbitrary structures is limited by the need to redesign the components for each new target structure. In Chapter 1 we described an alternative dynamically programmable fluidic assembly method which allows the assembly of arbitrary structures from regular components while avoiding the limitations of direct-manipulation techniques.

In this chapter we present the design and testing of mechanical and electrical interfaces for the micro-scale components assembled using this method. We first discuss the design and testing of mechanical latches used to hold assemblies together. We then discuss the fabrication and testing of an electrical interface to allow the transfer of power and/or communication between tiles. These advances greatly expand the possibilities of our dynamically programmable fluidic assembly concept.

Mechanical Interface

Our goal was to design a planar passive latching mechanism to hold assembled tiles together. It was important that the force required to latch two tiles together was within the range of achievable hydrodynamic forces in our system. We thus used a model based on elementary beam mechanics to relate the latch design parameters to the force required to engage the latches.

Design

Figure 2.1 is a schematic of a latch with relevant design parameters and forces indicated. In our model we assume that the latch is a fixed, planar cantilever beam. We further assume that this beam is deflected by a complementary rigid shape which is constrained in the y-direction (due to pairing of opposing latches). This model also

assumes no friction between the latch and complementary surfaces.

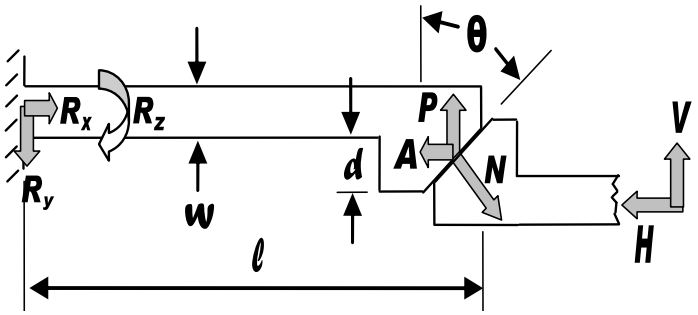


Figure 2.1. Latch parameters. A common set of latch parameters were chosen to compare various latch designs. The maximum required lateral force H on a contacting structure required to bend a latch it's full travel d was calculated for each design. l is the length of the latch, w is the width, t (in the z-direction) is the thickness, and θ is the angle of the latch head.

The perpendicular force *P* required to bend the cantilever tip by a deflection *d* is given by $P = 3dEI / l^3$ where *E* is the modulus of elasticity (we used 1.50×10^{11} Pa for silicon), *I* is the area moment of inertia of the beam's cross section, and *l* is the length of the beam. The area moment of inertia for a square beam is $I = w^3 t / 12$ where *w* is the width of the beam and ***t*** is its thickness. *P* can be related to the force ***N*** normal to the latch head by the trigonometric relationship $P = N \sin\theta$, similarly *N* is related to the horizontal force *H* by the relationship $H = N \cos\theta$. Finally, the total latching force *F* is this horizontal force multiplied by *n*, the total number of latches deflected per side $F = nH$. Putting this all together, the latching force in terms of the latch parameters is given by the equation:

$$F = \frac{ndw^3 tE}{4l^3 \tan\theta} \tag{2.1}$$

Various tile designs (Figure 2.2) were created by varying the parameters in Equation 2.1 while the overall shape and size of the tiles were held constant. Table 2.1 summarizes the parameters chosen for each design. Note that the dimensions are the

nominal design dimensions chosen so two tiles mate exactly. During fabrication the tile patterns were slightly overexposed in order to leave a clearance between the latching parts. These fabrication biases were measured from SEM images of the fabricated tiles and included in Table 2.1. The predicted latching force for each tile design was then calculated taking these biases into account.

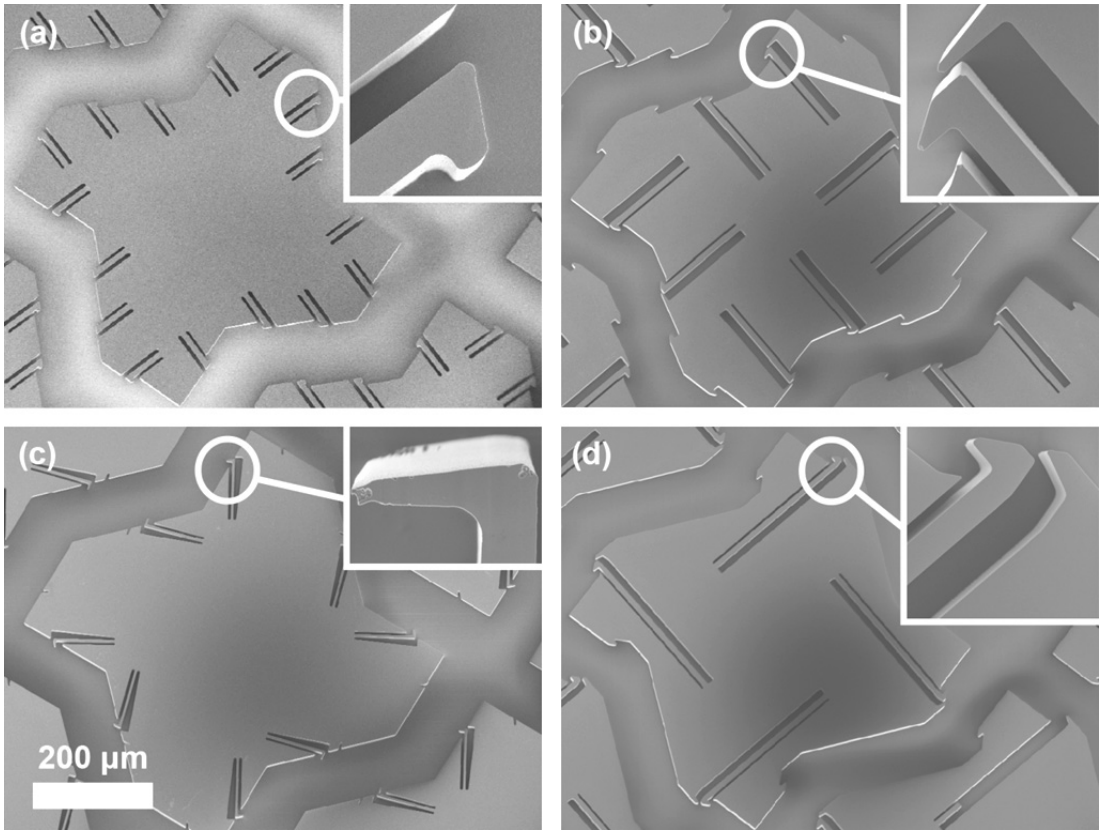


Figure 2.2. Latching microtile designs. Scanning electron microscope images of latching 500 µm × 500 µm × 30 µm tiles with latch detail insets. Pictured are (a) design 1, (b) design 2, (c) design 3, and (d) design 4. Designs 5 and 6 (not pictured) are identical to Design 4 except with thinner latch shafts.

Fabrication

Batches of the six microtile designs were each fabricated from a silicon-on-insulator (SOI) wafer with a 30 µm device layers and 1 µm thick buried oxide. A 7 µm

layer of SPR 220 series photoresist was patterned on each wafer by contact alignment. The tile outlines were then etched through the device layer with a deep Bosch etch. The tiles were then released using a wet etch with 49% HF, and collected by filtration.

Table 2.1 Latch design parameters

Design	N	L (μm)	W (μm)	D (μm)	Bias (μm)	θ (°)	F (μN)
1	4	65.8	10.0	2.5	0.23	26.6	62498
2	4	158.0	10.0	5.0	0.31	34.0	6527
3	4	97.2	5.0	11.0	0.40	60.5	2176
4	2	279.7	10.0	10.0	0.58	39.8	803
5	2	280.3	5.0	6.0	0.45	49.4	33
6	2	280.3	3.0	6.0	0.45	49.4	5

Results

The efficacies of the six tile designs were tested experimentally. Designs 1 through 3 were found not to latch within the limit of attainable hydrodynamic forces in the microfluidic chamber. However, it was possible to latch these designs with direct manipulation [Figure 2.3(a)]. Design 4 latched permanently in the chamber into two- and three- tile assemblies [Figure 2.3(c) and (d)]. Designs 5 and 6 also latched together due to hydrodynamic pressure, however small latch travels and large fabrication biases [Figure 2.3(b)] caused these tiles to disassemble under shear force conditions.

The ability of microtiles with latch design 4 to assemble under achievable flow conditions within a microfluidic chamber makes it a good candidate for dynamically programmable fluidic assembly experiments. Furthermore, based on the latching forces calculated in Table 2.1, we conclude that the maximum hydrodynamic forces applied to the tiles under normal flow conditions are on the order of 800 μN.

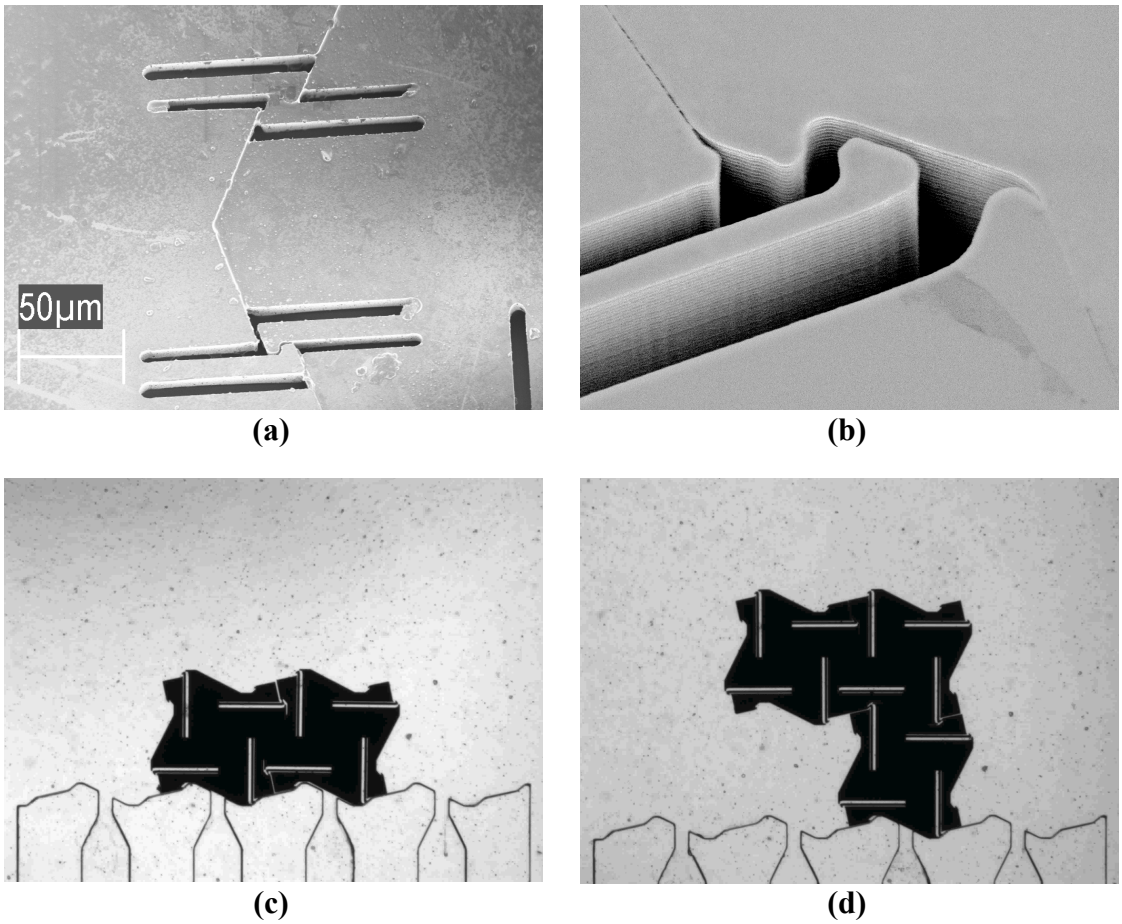


Figure 2.3. Latched Microtiles (a) SEM of two manually latched Design 1 tiles. (b) SEM of latched Design 6 microtiles. A large fabrication bias causes poor mating. (c), (d) Frames from videos of the assembly of two and three Design 4 tiles (respectively).

Electrical Interface

The goal of the electrical interface design was to develop a method of establishing electrical contact between fluidically assembled MEMS components at their edges. Such interfacing is important in extending our dynamically programmable fluidic assembly concept to the assembly of active devices. To date, electrical connections between fluidically assembled components have only been established between their larger, horizontal surfaces using surface-energy minimization techniques.

Design

In order to establish electrical connection between tiles, basic electrodes were patterned on the tiles connecting to contact pads on each tile side (Figure 2.4). This design takes advantage of the well-defined mechanical interface in order to line up the electrical contacts and hold them together to form an electrical connection. Obviously tiles in this case would have to be assembled within a non-conducting fluid to prevent short-circuits. [Note that we have previously demonstrated assembly of silicon tiles in silicone oil, as discussed in Chapter 1 and (Tolley, Zykov, Erickson, & Lipson, 2006)].

Fabrication

Electrodes were patterned Design 5 latching tiles etched through the device layer of an SOI wafer (but not released) as described above. We first deposited 15 μm of chromium (for adhesion), followed and 80 μm of gold using electron gun evaporation. For uniform metal deposition on the tile tops and sidewalls, the wafer was held at 45° to the evaporation material, and rotated in plane during deposition.

A thick layer (40 μm) of AZ-4903 photoresist was used to protect the electrode layer on the surface and sidewalls of tiles during a wet etch of the gold and chromium. The thick photoresist required a two-step expose/develop sequence. The first step included exposure and under-development of the desired electrode pattern on the top of the tiles, while the second step included a long exposure with a different mask of the thick layer of photoresist between the tiles (excluding the contact pads). A final develop then removed the remaining resist [Figure 2.4(b)]. Tiles were released from the SOI wafer using a 49% HF oxide etch, which did not significantly damage electrode integrity.

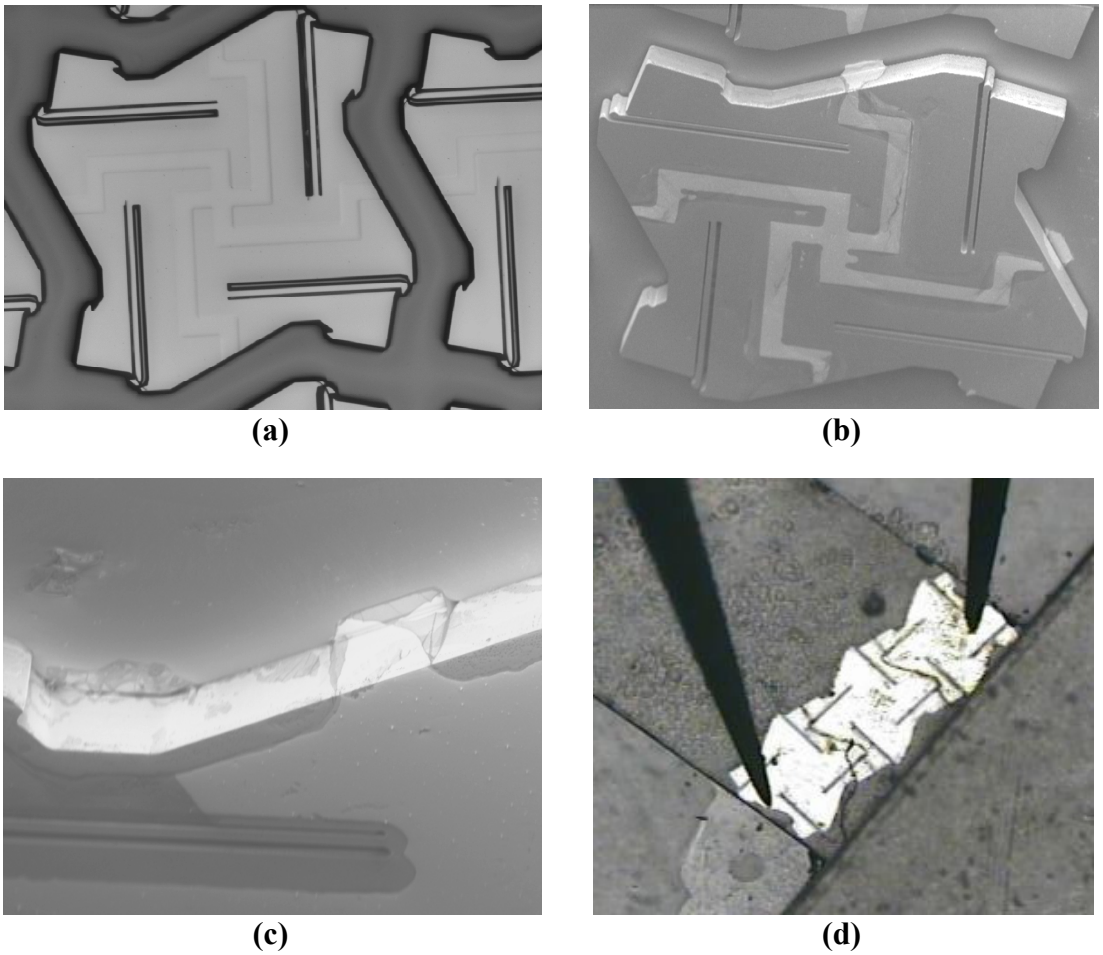


Figure 2.4. Images of patterned microtiles. (a) Optical Micrograph of resist coated tiles after first exposure step (b) Scanning Electron Microscopy (SEM) image of patterned chromium/gold electrodes on top and sides of 500µm square by 30µm microtiles. (c) SEM of contact pad electrode on tile sidewall allowing horizontal electrical connection between tiles. (d) Measurement of resistance across three microtiles using contact probes.

Results

Fabrication yielded many tiles with intact electrodes and contact pads [Figure 2.4(a) and (c)] although inconsistencies in photoresist coverage led to varying results. Electrical testing was completed to verify conductivity across tiles and between assembled microtiles [Figure 2.4(d)]. Resistance data was obtained using a multimeter probe station for one, two, and three-tile circuits [Figure 2.5(a)]. Tiles were

manipulated and connected in silicone oil on a glass substrate using a pair of multimeter probe tips. As a control, the resistance across plain (electrodeless) silicon tiles was also measured [Figure 2.5(b)].

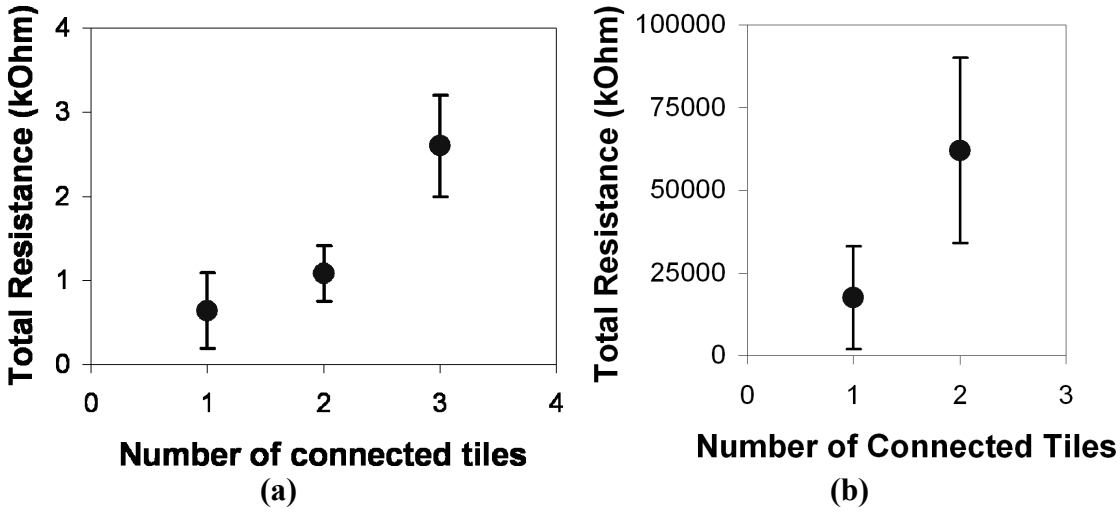


Figure 2.5. Resistance versus number of tiles for (a) electrode-patterned and (b) non-patterned silicon microtiles Data points represent average measured values. Error bars represent maximum and minimum measured values. These graphs show that conduction occurs through tiles connected in-plane through gold electrodes with resistance values on the order of 104x less than through silicon.

We used the resistance equation $R = \rho L / A$ to calculate the order of magnitude of the resistance *R* across the width of one silicon tile with gold and chromium electrodes on top. Using values of 2.4e-8, 13e-8, and 1.0e-3 Ω-m as the resistances of gold, chromium, and silicon at 20°C respectively, we calculated $R \approx 4.0\Omega$. Compared with the resistances measured across a single patterned tile [Figure 2.5(a)], we conclude that *R* is negligible compared to contact resistance at the tile-probe and tile-tile interfaces. Under this assumption, using a least-squares regression, the average contact resistances between tiles and at the tile-probe interfaces were 880 Ω (0.00792 $\Omega \cdot cm^2$) and 280 Ω respectively.

Conclusions

We have presented the design and implementation of electrical and mechanical interfaces for fluidically assembled planar MEMS components. We have demonstrated the ability of components with these interfaces to bond by way of passive mechanical latches and establish an electrical connection. This work forms an integral part of the development of a novel microassembly strategy which bridges the gap between bottom-up self-assembly and top-down direct-manipulation techniques. Future work in this direction could lead to MEMS devices capable of the on-demand self-assembly, repair, and reconfiguration that lead to the adaptability and robustness of biological systems.

CHAPTER 3: HIERARCHICAL FLUIDIC MICROASSEMBLY FOR FAB-ON-A-CHIP SYSTEMS

Abstract

Developments in the miniaturization of fluidics have led to the emergence of many successful lab-on-a-chip systems that perform the functions of a chemical or biological lab on a small microfluidic chip. Inspired by this, we seek to develop *fab-on-a-chip* systems capable of assembling entire functional devices on a credit card-sized chip. Our approach is based on our previously described dynamically programmable microfluidic assembly process. In this chapter we identify the capability of hierarchical assembly as a key component of a fab-on-a-chip system. We then present two sets of experiments aimed at demonstrating hierarchical assembly on a microfluidic chip. The first set of experiments demonstrates hierarchical assembly using the experimental system described in Chapter 1. The second set demonstrates the functionality of a new system specifically designed for hierarchical assembly. We discuss the elements of this new system that were successful and some challenges that remain to be addressed.

Introduction

Over the past half century, technological revolutions caused by miniaturization science have spread from electronics to mechanical, fluidic, chemical, and bioanalysis systems. In all these areas, miniaturized systems offer many advantages over their macroscale predecessors such as improved efficiency, parallelizability, and cost, often in addition to enabling new functionalities. Lab-on-a-chip and micro total analysis systems (μTAS), for example, integrate traditional chemical or biological laboratory functions onto a microfluidic chip with an area of a few square centimetres (Manz,

1990; Stone, Stroock, & Ajdari, 2004; Thorsen, Maerkl, & Quake, 2002; Vilkner, Janasek, & Manz, 2004). Inspired by such work, we aim to extend the benefits of miniaturization to the area of microdevice fabrication, with the goal of creating a general fab-on-a-chip assembly system. Figure 3.1 illustrates our concept of a handheld assembler capable of fabricating a microdevice (e.g. a chemical detector) on demand in an unstructured environment (e.g. the site of a natural disaster). Another example application is the assembly of custom implantable devices for disease monitoring and/or therapy.

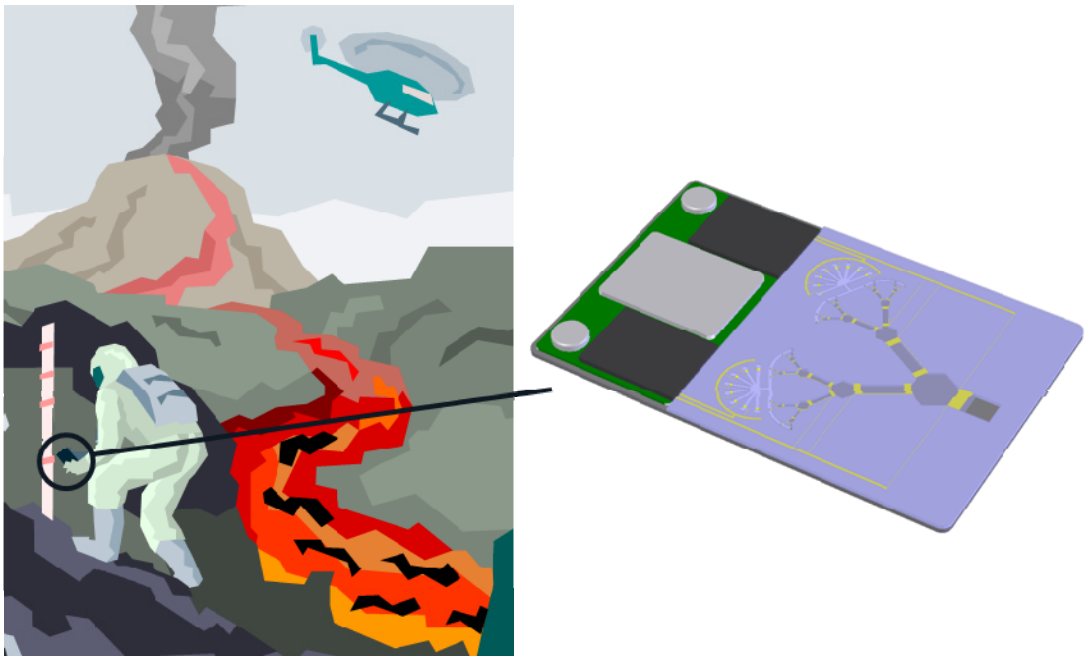


Figure 3.1. Fab-on-a-chip concept. A fab-on-a-chip system allows the assembly of custom, inexpensive microdevices on very small platforms in unstructured environments.

Chapter 1 gives a review of previous work in micro- and nanoscale assembly. Generally, these approaches rely on self-assembly forces to fabricate structures from components that would be very difficult to manipulate directly. However, self-assembly relies on the fact that the components are pre-programmed with the target

structure, in the way DNA contains the program for building an organism. This program may be simple, such as a hydrophobic pattern on a surface, but it is generally difficult or impossible to change. By definition, the assembly plan is built in to all of the individual components, thus each of the components must be changed or replaced in order to change the target assembly. A general fab-on-a-chip assembly device, however, should be programmable on-the-fly. It should contain all of the capabilities necessary to change the assembly program to fabricate any specified structure. Thus, self-assembly alone does not form a sufficient basis for a general fab-on-a-chip device. It must have either the capability to create pre-programmed components for self-assembly, or a means of programming the assembly process.

In this work, we take the latter approach. Based on the work presented in Chapter 1, we pursue the fab-on-a-chip concept using dynamically programmable fluidic assembly. Fluidic manipulation is used (as in nature) since stiction and friction become overwhelming burdens to dry assembly at small scales (Böhringer, Fearing, & Goldberg, 1999). The biggest drawback with the approach presented in Chapter 1 for fab-on-a-chip is the slow rate of assembly. Since assembly was performed serially (i.e. one component at a time), the fabrication of a structure of an appreciable size relative to the size of the components would take a prohibitively long time. Additionally, an irreversible failure during assembly results in a completely unusable assembly and the process must restart from the beginning.

Hierarchical Assembly

In order to addresses these issues of serial assembly, this work presents an alternative approach: hierarchical assembly. Recent work in nanofabrication has identified hierarchical assembly as a promising approach for the self-assembly of nanoscale devices (Huang, Duan, Wei, & Lieber, 2001; Jenekhe & Chen, 1999; Yang,

et al., 1998; Lopes & Jaeger, 2001; Whang, Jin, Wu, & Lieber, 2003). Here, we propose to use a hierarchical approach to dynamically programmable assembly. Figure 3.2 illustrates this concept. The target structure is fabricated by dividing the work up among many small assembly facilities that manufacture subassemblies from raw components. These subassemblies are then transported to larger facilities that use them to fabricate larger subassemblies. This process continues until the final structure is completed.

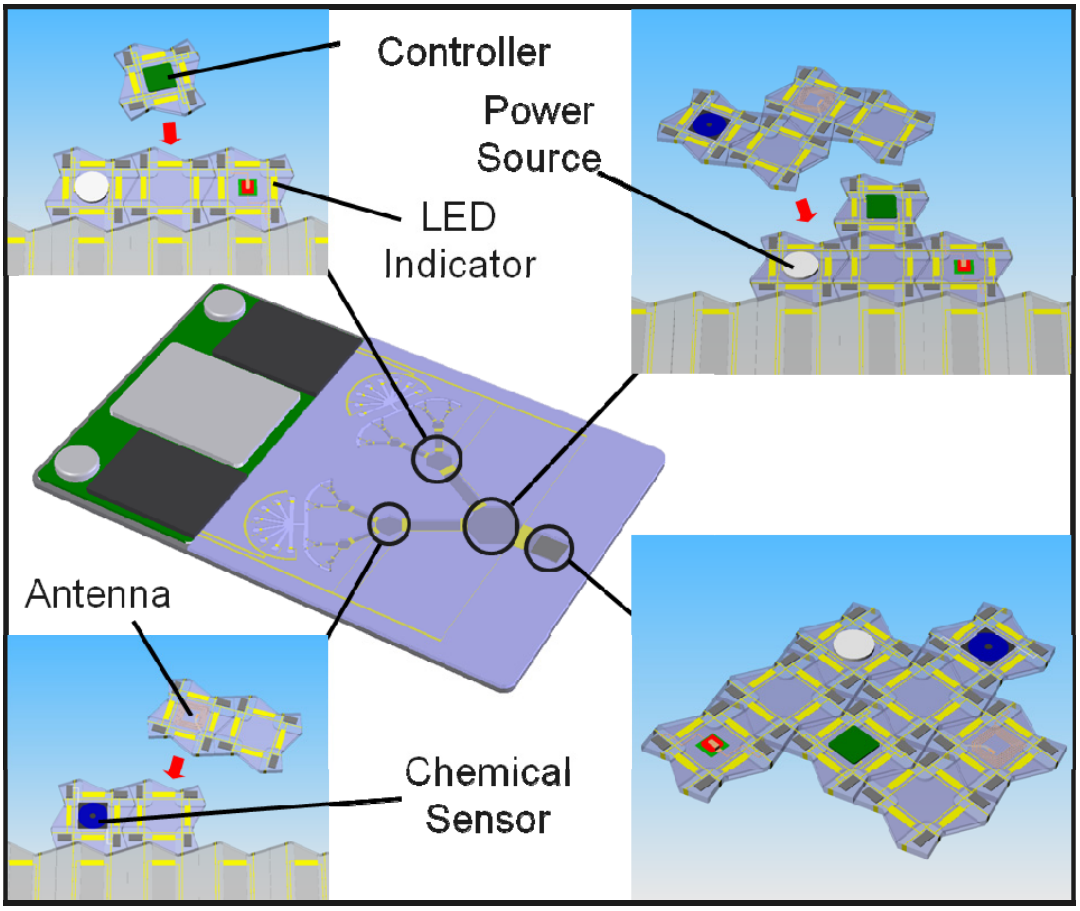
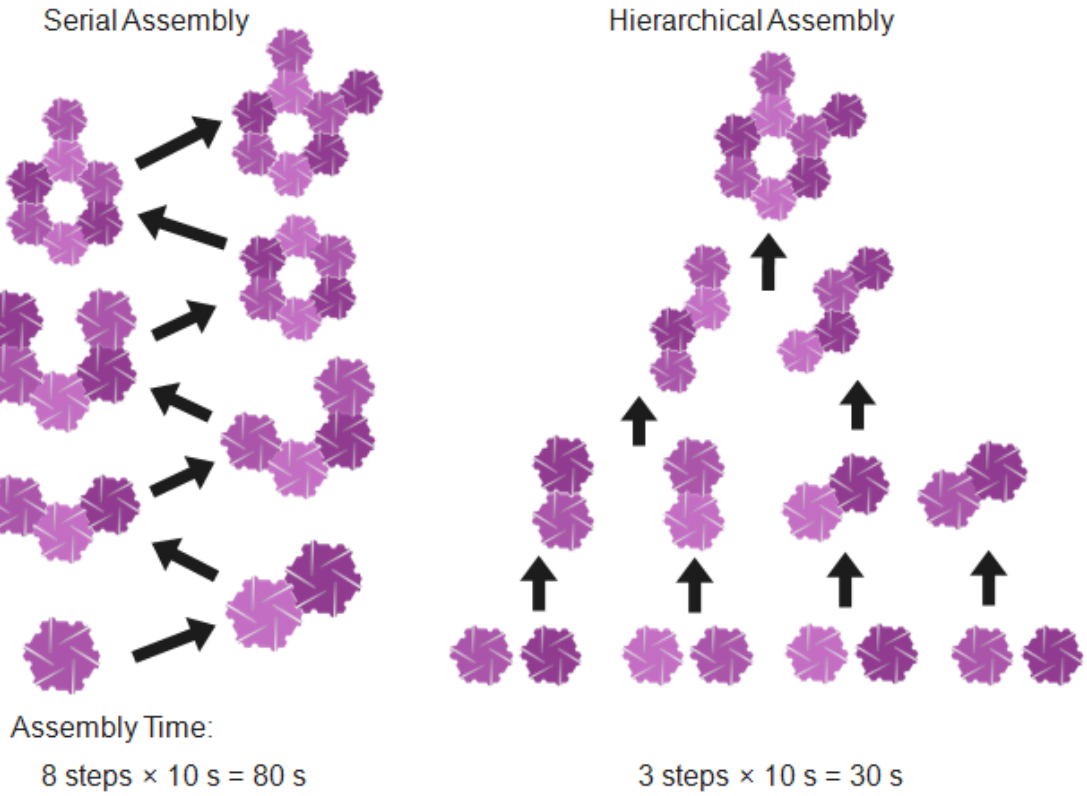


Figure 3.2. Hierarchical assembly for fab-on-a-chip. Programmable fluidic microassembly is used to assemble functional devices hierarchically from heterogeneous components. Subassemblies are fabricated in parallel to reduce time and the cost of assembly errors. A hierarchical system manipulates increasingly larger subassemblies to form the final device.

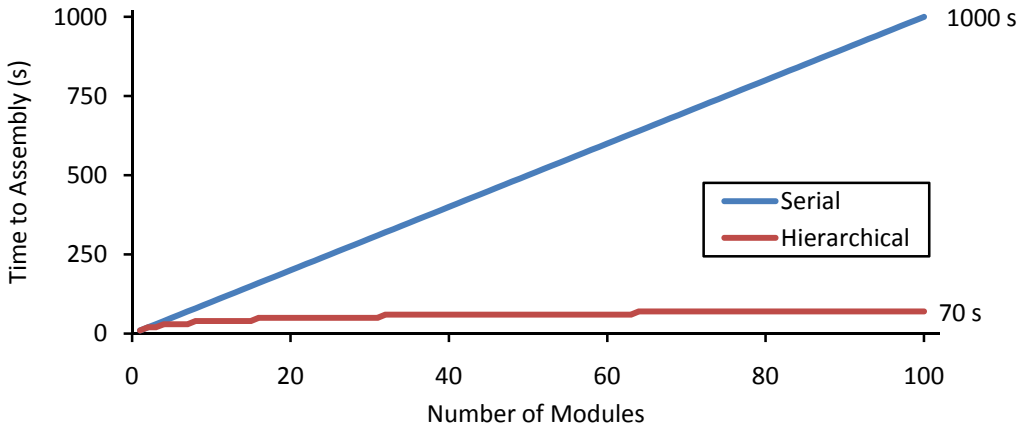
The two primary advantages of this approach are its reduction of assembly time and the cost of errors. Assembly time is reduced by the parallel assembly of many components. This effectively reduces the assembly time from $O(n)$ (i.e. linear) in the number of components to $O(\log(n))$, where the base of the logarithm is determined by the branching factor of the hierarchy. Figure 3.3(a) illustrates this concept for the assembly of an eight-module structure. The plot of Figure 3.3(b) compares the time required for serial and hierarchical assembly of structures composed of up to 100 modules. This plot assumes a branching factor of two per level of hierarchy and a time cost of 10 s per assembly. The total assembly time for a 100 module structure is almost 17 min in the serial case and just over 1 min in the hierarchical case.

The second main advantage of the hierarchical approach is the reduction in the cost of errors. This is due to the fact that any irreversible error only renders the sub-assembly unusable (as opposed to the entire assembly in the serial case). Along the same lines, errors with the assembly facilities themselves are more easily tolerated due to their large redundancy.

Of course, these advantages come with costs and challenges of their own. A hierarchical microfluidic assembly device as we have described would clearly be much more complicated than we presented in Chapter 1. In this chapter we present two sets of results aimed at addressing some of these challenges. In the next section we present a set of experiments that demonstrate the concept of hierarchical microfluidic assembly in a single chamber. These experiments used the square components originally presented in Chapter 1. The following section describes a new, multilayer chip with multiple chambers designed to investigate some of the challenges of a hierarchical infrastructure. There we present results from experiments in the manipulation and assembly of 100 μm side-length hexagonal tiles.



(a)



(b)

Figure 3.3. Reduced time cost of hierarchical assembly. (a) Comparison of serial and hierarchical approaches to the assembly of an eight module structure. (b) Plot comparing serial and hierarchical assembly times versus structure size. The time to assemble a structure scales linearly with the number of modules while the time for hierarchical assembly scales logarithmically.

Hierarchical Microfluidic Assembly

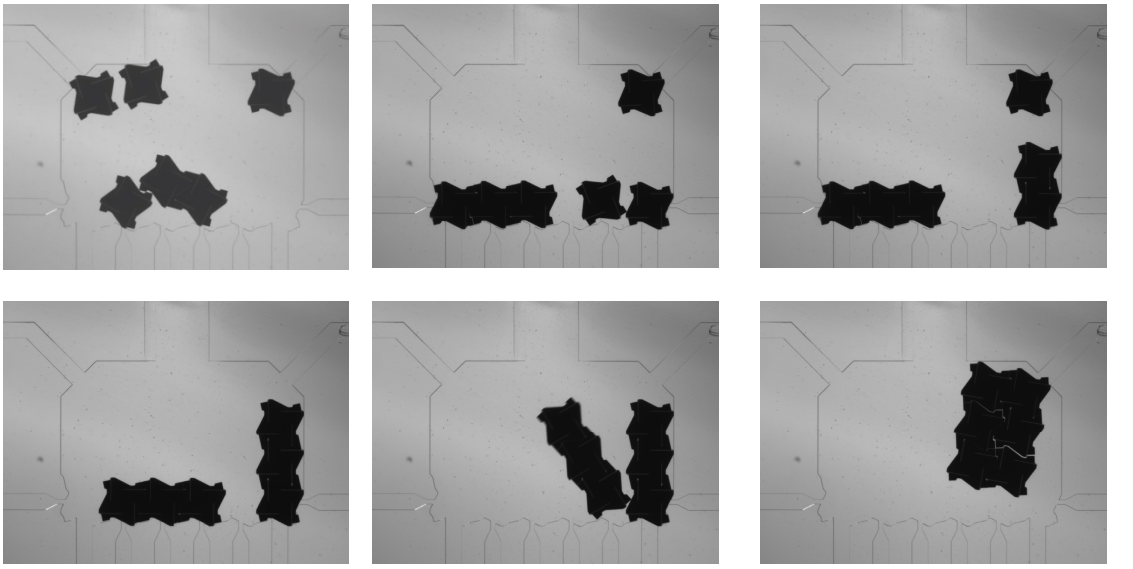
This section describes experiments that demonstrate hierarchical microfluidic assembly. These experiments are performed using the microfluidic chip and square tile designs originally presented in Chapter 1. In order to demonstrate the concept of hierarchical microfluidic assembly, these experiments involved the fabrication of separate sub-assemblies that were subsequently joined to form the final structure. Two of these experiments demonstrate the assembly of six-module structures from three-module parts while the third demonstrates three stages of hierarchical assembly with a final structure composed of eight modules. In order to demonstrate that the assembly was not random, all of the subassemblies were identical although this is not required for hierarchical assembly.

Materials and Methods

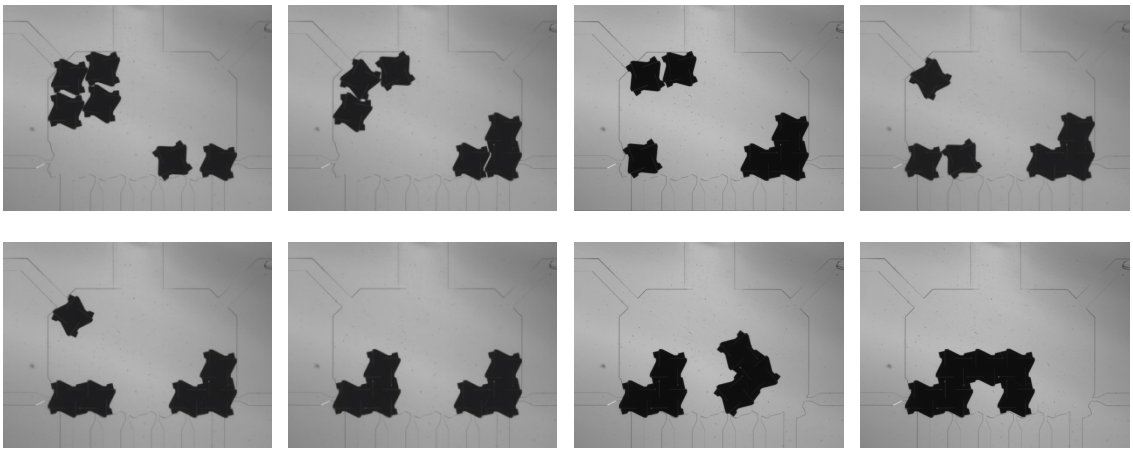
The microfluidic chip and tile design and fabrication for the experiments presented in this section are identical to those originally presented in Chapter 1. Thus, we refer the reader to the *Materials and Methods* section of Chapter 1 for details. Tile manipulation was also achieved in a similar method to the previous experiments. All of the fluidic valving in these experiments was human-controlled with visual feedback although in principle it could be done automatically.

Results

The first pair of experiments demonstrates the concept of hierarchical assembly in two stages. The first experiment separately assembles two lines of three tiles each, which are then assembled in to a six-tile rectangle [Figure 3.4(a)]. The second assembles two L-shapes, which were then united to form a more complex structure [Figure 3.4(b)].



(a)



(b)

Figure 3.4. Hierarchical assembly experiments. Two frame sequences from video micrographs of hierarchical assembly experiments. (a) Assembly of a rectangle from two three-tile lines. (b) Assembly of two L-shapes into a larger structure.

The goal of the third experiment was to demonstrate three-stage hierarchical assembly. For the first stage, four cubes were assembled into two pairs. These pairs were then united to form a square for the second stage [Figure 3.5(a)]. This process was then repeated to form a second square (due to space constraints the second square could not be assembled in parallel). For the third stage of hierarchy, the two four-cube squares were united to form an eight-cube rectangle [Figure 3.5(b)].

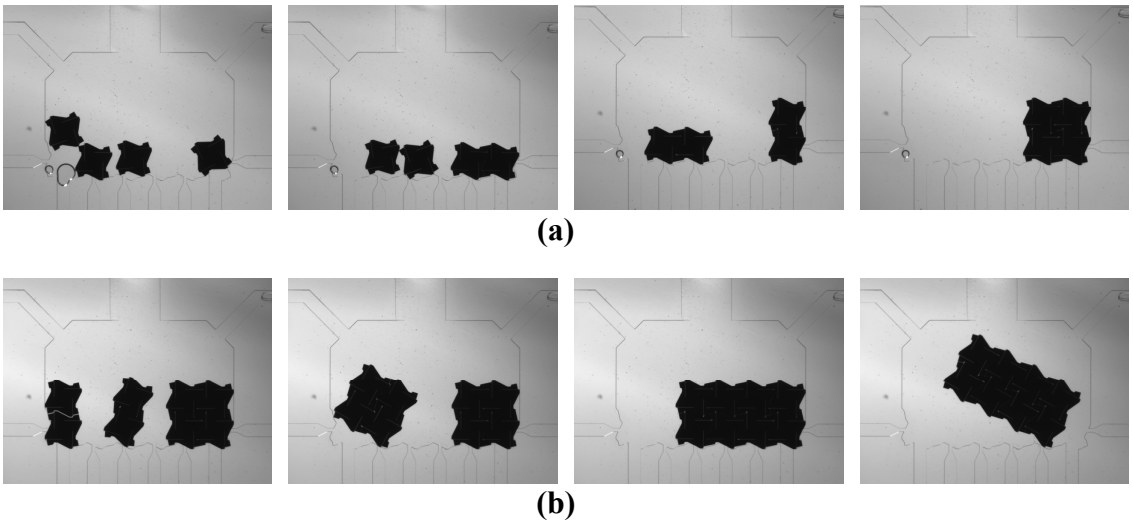


Figure 3.5. Three-stage hierarchical assembly. Frame sequence from a video micrograph of a three-stage hierarchical assembly experiment. (a) Four individual modules are first assembled into tile pairs that are then assembled into a square. (b) Two squares are then assembled into an eight-tile rectangle to form the third stage of hierarchical assembly.

Discussion

The three experiments presented in this section demonstrate our concept of hierarchical assembly. However, there are some limitations to these experiments. Since these experiments were performed in a confined assembly chamber, the overall size of the assemblies was limited. Consequently, the number of attainable levels of hierarchy was also limited. Similarly, the space constraints, as well as the limited abilities of a single operator, prevent the demonstration of parallel assembly since the substructures had to be assembled one after another. Additionally, some of the substructures had to be assembled in different orientations, following different assembly sequences. Thus, although the subassemblies were fabricated separately, the process did not demonstrate the time-saving promise of hierarchical assembly.

We should mention, however, that we have previously demonstrated the

automated assembly of all of the first stage subassemblies in Chapter 1. Thus, combining that previous work with the methods used in the current experiments would in principle lead to experimental parallel hierarchical assembly. The next section describes a new microfluidic chip and tile design to address some of these limitations in demonstrating hierarchical assembly.

Multi-Chamber Hierarchical Microfluidic Assembly System

The experiments presented in the previous section demonstrated hierarchical assembly, although the assembly size and number of stages of hierarchy were limited by the small size of the assembly chamber relative to the size of the modules. Additionally, since the subassemblies had to coexist in the same chamber, the environment was different during each assembly. In this section we present a new multi-chamber microfluidic chip and accompanying tile design to address these issues and further demonstrate the Fab-on-a-Chip concept.

Microtiles

The microtile design for the multi-chamber hierarchical assembly chip is shown in Figure 3.6. The tiles themselves are smaller than those used in previous experiments. In terms of area, the new tiles are almost an order of magnitude smaller (0.026 mm^2 versus 0.25 mm^2). Their width is also 40% of the previous tiles (12 μm versus 30 μm). This smaller size allows many more tiles to fit onto a chip of a given size. Additionally, the size of these tiles is such that they can be easily sucked or pushed through the tip of syringe for storage or deployment without damage.

Another significant change with these tiles is their hexagonal shape. The new shape was chosen as a way to realize higher connectivity with smaller structures. For example, each pair of tiles is connected in the three-tile structure shown in Figure

3.6(d), unlike a three-tile structure composed of square modules. Nonetheless, the basic features of an alignment pattern with a single passive latch per side remain. The fabrication process for these tiles is exactly the same as that for the square tiles described in Chapter 1, except that the device layer on the original SOI wafer is 12 µm thick instead of 30 µm.

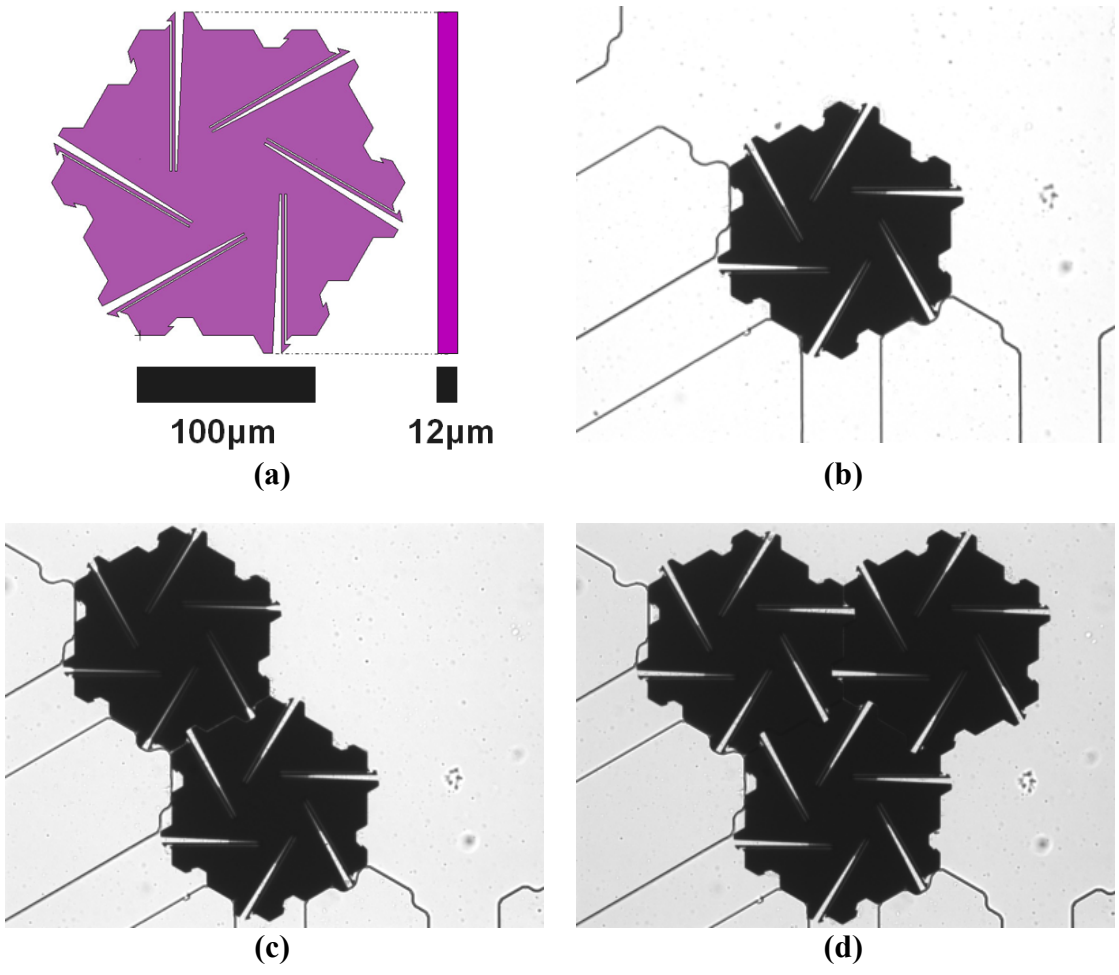


Figure 3.6. Hexagonal microtile design. As with the square design, hexagonal tiles have a single compliant latch per side. (a) Design schematic with dimensions indicated. (b) Micrograph of an actual tile in a microfluidic chip. (c)–(d) Micrographs of assemblies of two and three tiles (respectively), demonstrating hexagonal assembly scheme.

Chip Design

The design of the multi-chamber hierarchical microfluidic assembly chip is shown in Figure 3.7(a). The fluidic layer consists of three hexagonal chambers for the assembly and manipulation of the aforementioned silicon tiles. The centre chamber is similar to the assembly chamber of the fluidic chip from Chapter 1. It has a substrate along the bottom where a number of fluid channels connect to the chamber. The substrate has a pattern that compliments the tile alignment patterns such that one tile sits adjacent to the opening of each channel [see Figure 3.6(b)–(d)]. The exception to this is two corner locations on the substrate where two channels come together to one tile location. On the top of the chamber, another channel is used to inject flow during assembly.

Two larger channels connect the central chamber to two additional chambers. The chamber on the left is used to store tiles for assembly and the chamber on the right is used to store sub-assemblies during hierarchical assembly. To the left of the left chamber is a larger channel that connects to a port used for the introduction of tiles to into the chip. Similarly, a large channel to the right of the right chamber is used to extract tiles from the chip. A fluid channel connected to the top of each of these chambers facilitates the movement of tiles and assemblies between the three chambers.

Above the fluidic layer sits a pneumatic layer, coloured green in Figure 3.7(a). This pneumatic layer serves to close off the fluidic channels underneath according to the multilayered valving principle illustrated in Figure 3.7(b) (Unger, Chou, Thorsen, Scherer, & Quake, 2000). The pneumatic valving allows for controlled manipulation of tiles and assemblies between the chambers. It also acts as a barrier to keep tiles in the main chamber during assembly, and prevent others from entering. Figure 3.7(c) is a collation of micrographs of a multilayer microfluidic chip of this design. This chip

was fabricated by casting PDMS over a photoresist master, as described in the *Chamber Fabrication* section of Chapter 1.

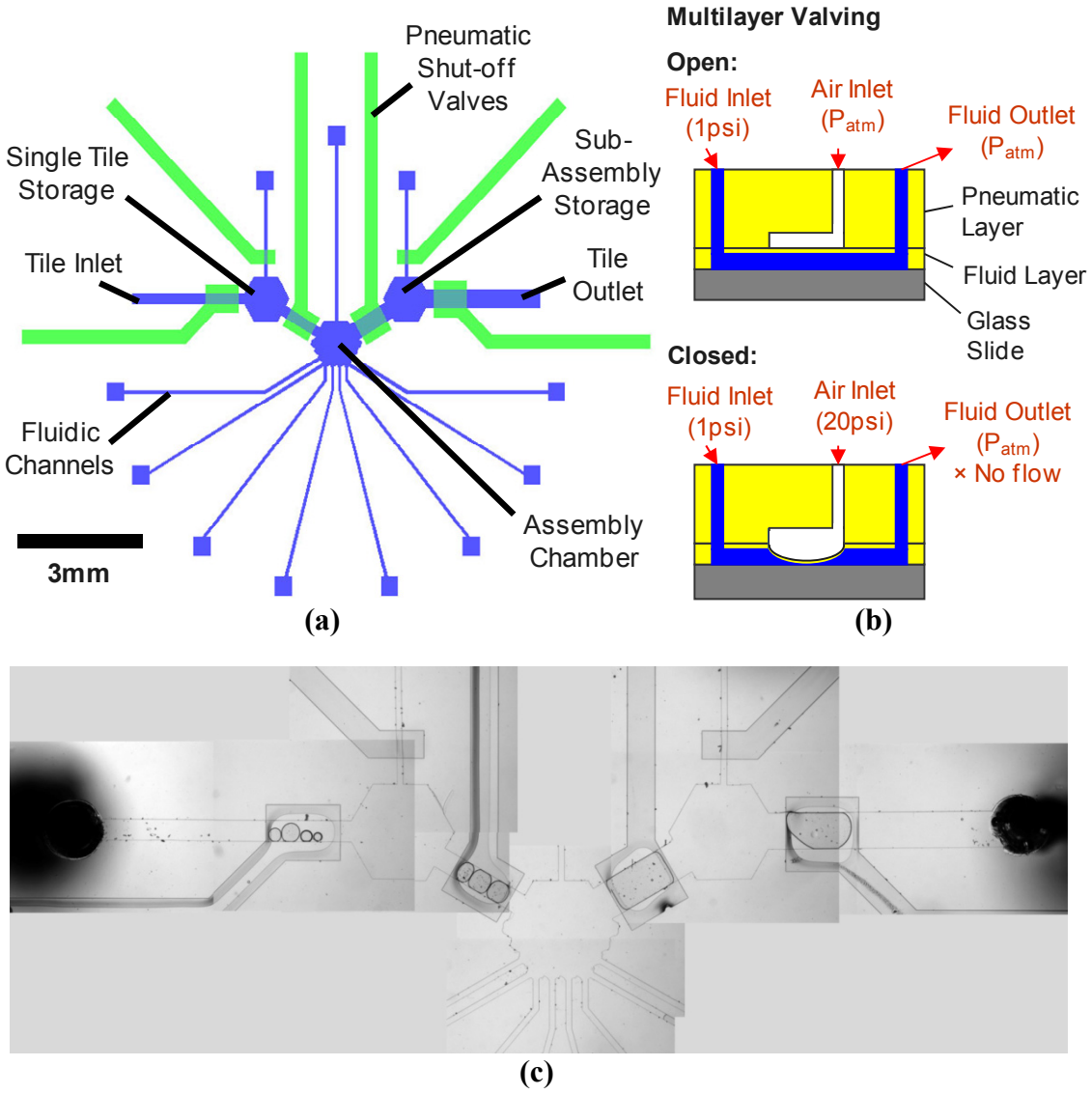


Figure 3.7. Multi-Chamber hierarchical microfluidic assembly chip. (a) Multilayer PDMS chip design allows on-chip valving to isolate three separate chambers. The left chamber is used to sort and store tiles to be assembled in the centre main assembly chamber. The right chamber is used to store sub-assemblies for hierarchical assembly. (b) Schematic of multilayer PDMS valving principle. Pressure is applied to the pneumatic layer to valve the fluid layer underneath. (c) Assembled micrographs of a physical microfluidic chip.

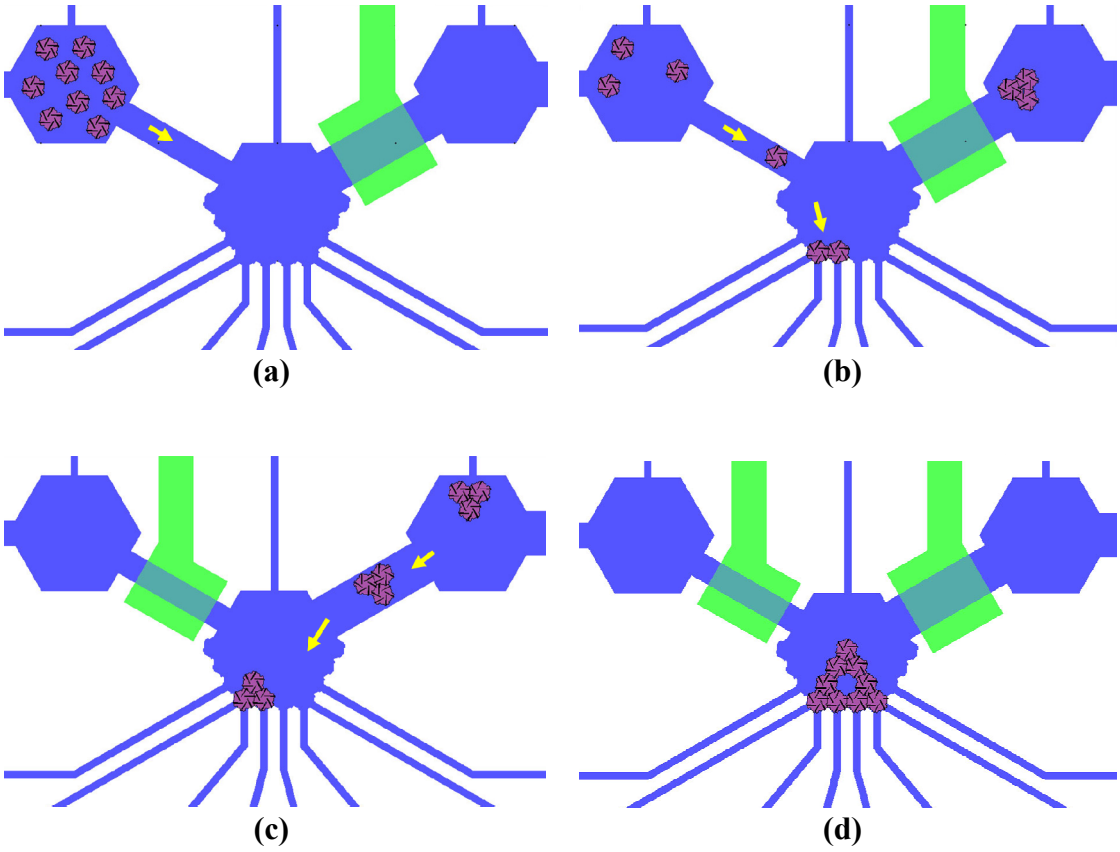


Figure 3.8. Hierarchical assembly procedure. (a)–(b) Microtiles stored initially in left chamber are moved individually to centre chamber for assembly. (c)–(d) Hierarchical assembly is achieved by storing sub-structures in right chamber while others are completed. They are then moved back to the centre chamber for the completion of the final structure. This process simulates the Fab-on-a-Chip concept where multiple chambers assemble sub-structures in parallel.

The operating principle underlying the chip design described in this section is illustrated in Figure 3.8. First, a set of tiles is collected in the left chamber by sorting and removing broken, defective, or inverted tiles. The experiment then starts by introducing these tiles one at a time into the central assembly chamber. During these assembly steps, the pneumatic valves are closed to prevent additional tiles from entering the assembly chamber. Once the target structure for this stage of assembly (a three-tile triangle in Figure 3.8) is complete, it is moved to the right chamber for storage. This allows the independent assembly of multiple substructures. In a fab-on-a-

chip system, these assemblies would happen in parallel. However for hierarchical assembly it is sufficient that they be assembled separately. Finally, when all of the substructures are complete, they are reintroduced one at a time into the main chamber to be assembled into the final structure.

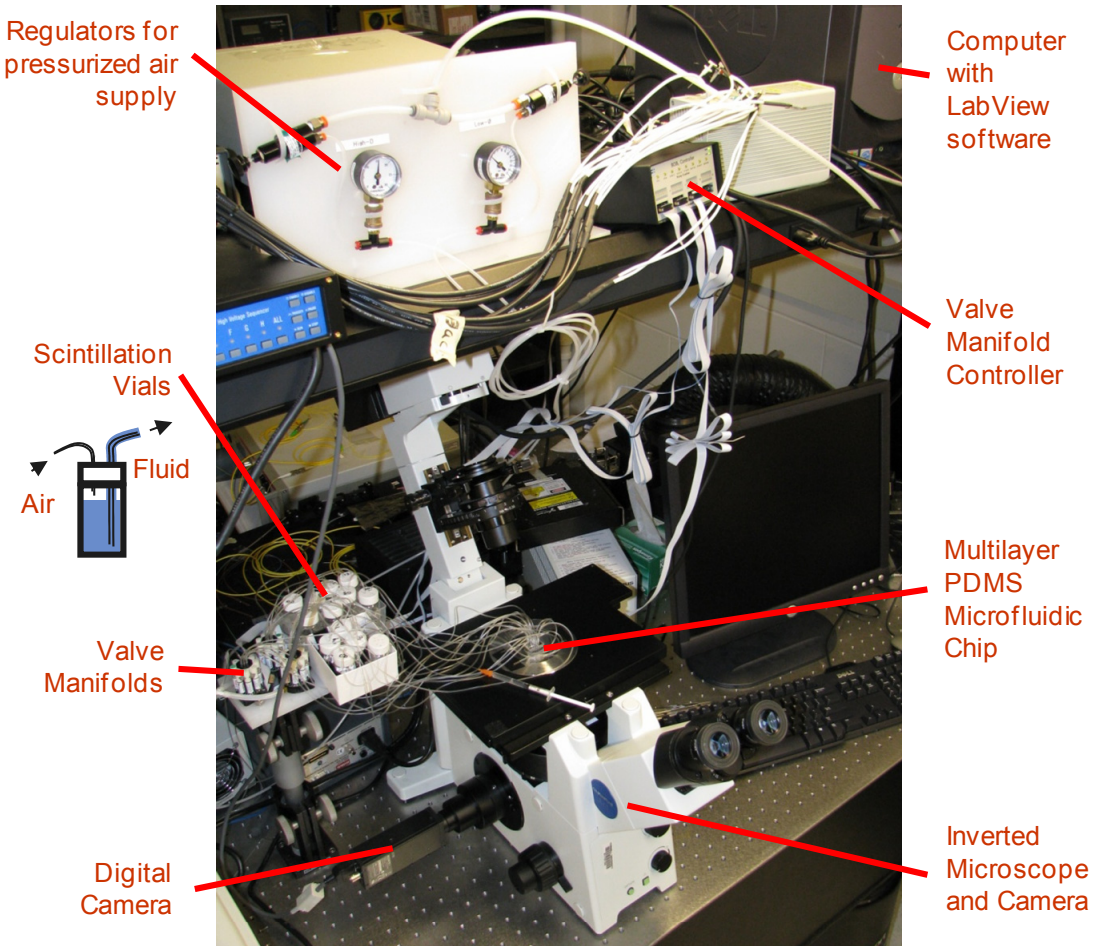


Figure 3.9. Experimental Setup. Photograph of experimental hierarchical microfluidic assembly setup with primary components indicated.

Experimental Setup

Our experimental setup for the multi-chamber microfluidic assembly system is shown in Figure 3.9 with the primary components indicated. The chip itself is

mounted on an inverted microscope with a digital camera for recording experiments. The valving of the chip's pneumatic layers is controlled by pressurized air lines passing through the valve manifolds. Fluid pressure is also controlled by valved pressurized air pumped into a set of scintillation vials. (The operating principle of the scintillation vials is shown in the schematic on Figure 3.9.) The manifold valves are operated through a controller with a PC using the LabView software. 17 valves are needed in all, 8 for the pneumatic layer and 11 for the fluidic layer.

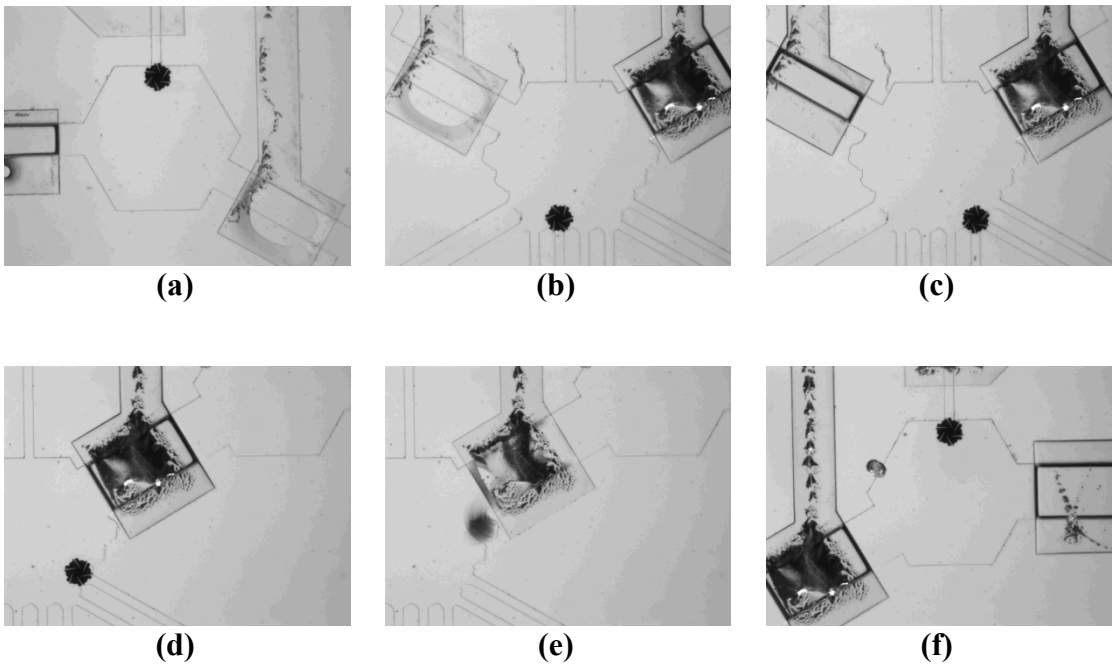


Figure 3.10. Demonstration of multi-chamber chip. (a) Pneumatic valves (top and lower-right) are opened to inject flow that moves a hexagonal microtile from storage chamber to assembly chamber. (b)–(d) Once valves are closed to seal off the assembly chamber, the microtile can be manipulated. (e) Opening the right valve allows tiles or assemblies to be moved to the right chamber for storage.

Results

Figure 3.10 shows a series of frames taken from a video micrograph of an experiment demonstrating the operation of the multi-chamber chip. A single tile is

inserted onto the chip from a syringe, then moved into the left chamber. The pneumatic valving is then switched to connect the left and centre chambers, at which point the fluid flow moves the tile into the assembly chamber. The manipulation of the tile among the ports of the assembly chamber is demonstrated to be very effective, at which point the tile is deposited into the right chamber for storage.

Other experiments demonstrated the manipulation and assembly capabilities similar to those demonstrated with the previous system in Chapter 1. For example, Figure 3.11 shows a series of frames from a two-tile assembly experiment. The images in Figure 3.6(c),(d) come from similar experiments demonstrating the assembly of two and three tiles (respectively).

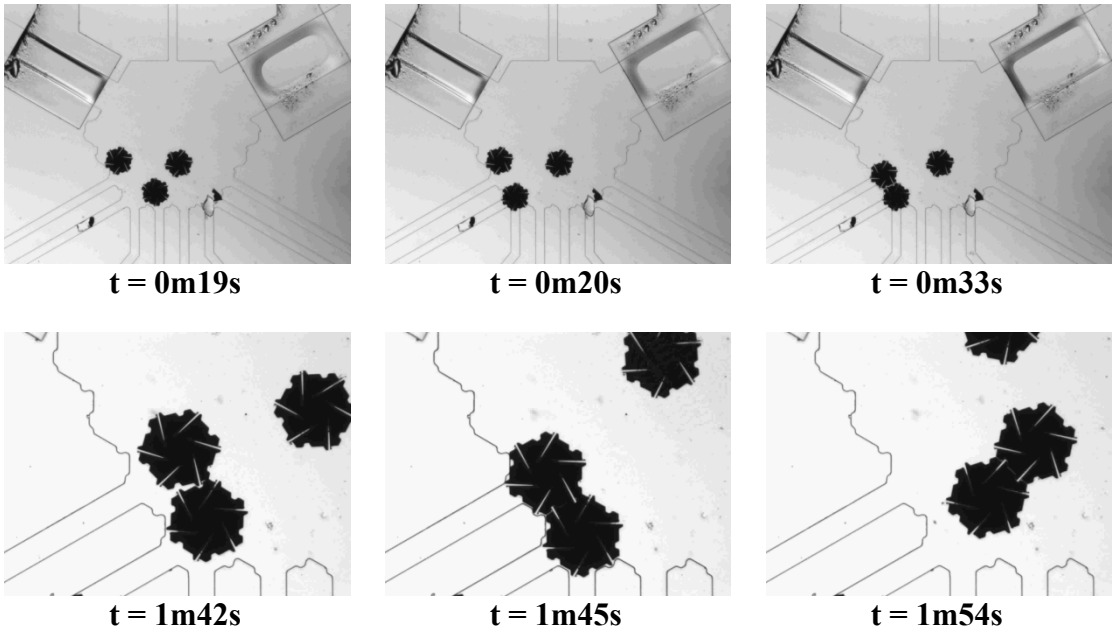


Figure 3.11. Two-tile assembly. Series of frames from a video micrograph of an experiment demonstrating the assembly of two hexagonal tiles. A higher magnification lens was used for the second half of the experiment corresponding to the images in the bottom row.

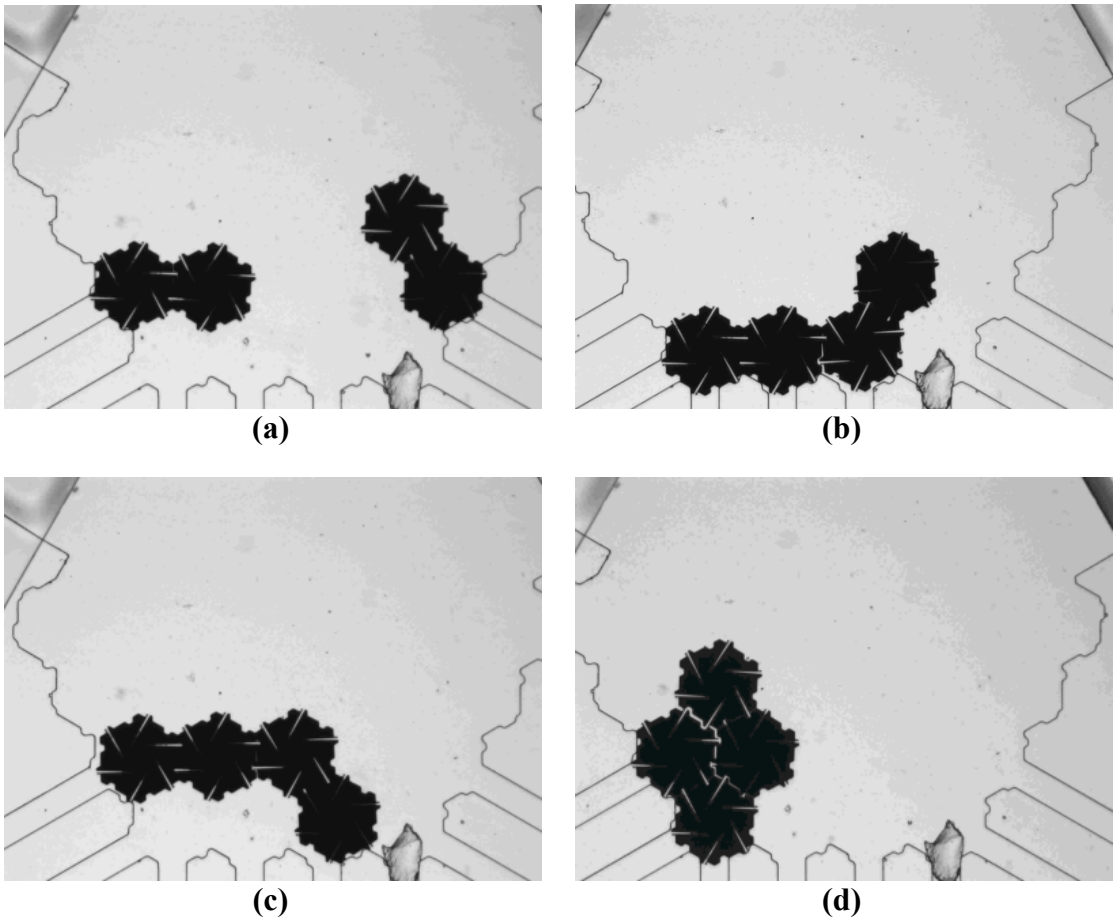


Figure 3.12. Hierarchical assembly experiments. Frames from experiments were conducted to demonstrate the hierarchical assembly of larger structures from smaller substructures. (a) Two assembled two-tile substructures. (b)–(d) Three different attempts at four-tile structure assemblies.

Figure 3.12 shows frames from experiments with the main goal of the multi-chamber hierarchical microfluidic assembly system: hierarchical assembly. Using procedures similar to that seen in Figure 3.11, we assembled two separate pairs of tiles [Figure 3.12(a)]. We then attempted to assemble these pairs into various larger structures to demonstrate hierarchical assembly [Figure 3.12(b)–(d)].

We were able to demonstrate the manipulation of the subassemblies into the various configurations shown in Figure 3.12. However, most of these structures did not latch completely in such a way that they would remain assembled during further

manipulations. The exception is the diamond shape in subfigure (d). This structure did remain assembled during manipulations despite the fact that only two of the six latches between the two subassemblies engaged completely. We attempted a number of solutions to induce the hexagonal tiles to latch such as increased pressure, pulsing the fluid flow directly and indirectly (by switching the pneumatic valves on and off rapidly), and external vibration with a small pager vibrator. However, we found none of these methods to be effective in inducing the tiles to latch.

Discussion

Based on the experiments described in the previous section, we found that the new microfluidic chip and tile design described in this section did achieve a number of its intended functions. We demonstrated the manipulation of tiles among the three chambers of the chip, sorting of broken and flipped tiles, and assembly of two- and three-tile structures. The smaller tiles did enable experiments on a smaller chip than the previous system. Also, their hexagonal shape led to more connected structures that stayed assembled very well during manipulation [such as the triangle structure shown in Figure 3.6(d)].

However, we did not achieve hierarchical assembly as envisioned in Figure 3.8. This was due to a variety of challenges, primarily with the new chip design. The main problem that plagued all experiments was a large amount of debris, most likely from the multilayer fabrication process [Figure 3.13(a)]. This problem was partially reduced by improved chip fabrication methods, but was never completely eliminated. The reason this problem has not been reported by others who have employed multilayer soft lithography techniques may have to do with the relatively large size of our channels and the increased pressures required to insert our microtiles into the microfluidic chip. Much of the debris seemed to enter our chips during these

operations [Figure 3.13(b)].

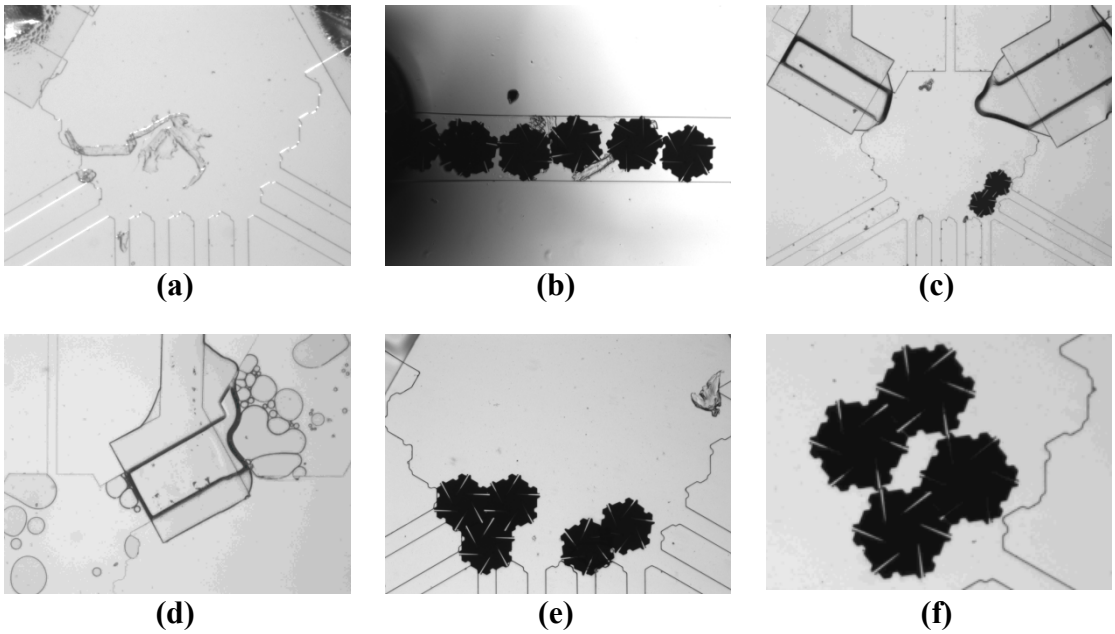


Figure 3.13. Multi-chamber chip challenges. Various challenges impeded the demonstration of hierarchical assembly on the multi-chamber chip. These include: (a),(b) Debris of unknown origin, (c),(d) delamination and valve leaks, (e) tile overlap in the chamber, and (f) non-specific tile binding.

Another constant source of problems was the pneumatic valving. In some cases, inadequate bonding between the pneumatic and fluidic layers led to delamination around the valves that eventually rendered the chip inoperable [Figure 3.13(c)]. Additionally, the effectiveness of the valves was very sensitive to the thickness of the layer between the fluidic and pneumatic layers. For large thicknesses, the layer was not sufficiently flexible for the valves to close off the flow in the channels underneath. However, thinner layers were susceptible to leakage, in which case air would enter the fluidic channels through the valves [Figure 3.13(d)].

Similar to the valving problem, the results were also highly sensitive to the depth of the fluidic channels. In many situations, the tiles would become stuck and seem to adhere to the top of the channels. The increased flow pressures required to dislodge

these tiles would often inflate the chambers to such a degree that tiles would become overlapped [Figure 3.13(e)].

Finally, an unanticipated problem was identified with the tile design during the hierarchical experiments. During attempts to assemble two pair structures, a particular interaction occurred that resulted in non-specific binding between the two subassemblies [Figure 3.13(f)]. Two pairs of latches bonded with each other (instead of with the intended complementary fixed parts). However, this situation occurred very infrequently and could probably be avoided with a small change to the design.

Conclusions

We have identified the capability of hierarchical assembly as an important feature of a practical programmable fluidic assembly system. Specifically, we envision it as a key component of a versatile "fab-on-a-chip" system. Inspired by this vision, we have conducted a number of experiments aimed at demonstrating hierarchical assembly in fluidic assembly systems. The first approach follows directly from our previous work on dynamically programmable fluidic assembly presented in Chapter 1. Using the same microtile and microfluidic chip designs from those previous experiments, we demonstrated two- and three-stage hierarchical assembly of structures composed of up to eight components.

In order to further demonstrate the hierarchical assembly of structures composed of more components from substructures assembled in identical environments, we developed a new microfluidic chip and accompanying microtile design. Our experiments with this new system demonstrated the basic functionality of the system and identified the hexagonal tiles shape as a viable alternative to the previously demonstrated square tiles. However, a number of unforeseen problems with the new system design prevented the demonstration of the envisioned hierarchical experiments.

We can only hope that future work will find solutions to these problems in order to realize functional fab-on-chip devices.

PART II: THREE DIMENSIONAL EXPERIMENTS

CHAPTER 4: THREE DIMENTIONAL STOCHASTIC FLUIDIC ASSEMBLY OF MINIMALISTIC COMPONENTS

Abstract

One of the grand challenges of self-reconfiguring modular robotics is the assembly of a functional system from thousands of components. However, to date, only systems comprised of small numbers of modules have been demonstrated. One approach to scaling to large numbers of modules is to simplify module design by relieving the modules of the power, control, and actuation requirements typically required for locomotion. Assembly is accomplished by taking advantage of stochastic environmental motions to move the modules into place. Here we present an experimental system to assemble 3-D target structures stochastically from simple, 1.5 cm scaled components by manipulating the fluid flow in a 1.3 L tank. We describe experiments demonstrating a range of fundamental capabilities necessary for 3-D stochastic fluidic assembly, including module manipulation, assembly of planar and non-planar structures, the repair of damaged structures, and parallel hierarchical assembly. We also discuss the control approach used to achieve 3-D assembly and evaluate the reliability of this approach. For a chosen four-module structure, we assembled the target shape in 6 out of 10 consecutive experiments in an average of 442 s.

Introduction

Self-reconfiguring modular robots offer many potential advantages over traditional robotics including the abilities to adapt their morphology to a given task and self-

repair when damaged. The repeated use of a small set of units could also potentially lead to reduced costs due to economies of scale and re-usability. However, in order to realize these advantages, the system must be able to scale to large numbers of small modules. While this feat is accomplished routinely in natural systems, to date, experimental robotic systems have not demonstrated the assembly of more than approximately 50 modules. Existing experimental systems typically assemble using deterministic processes where modules move directly to their target positions. However, this puts severe power, control, and actuation demands on the modules, usually over and above those required to fulfill their role in the final assembly.

Taking inspiration from nature, we follow a stochastic approach to modular robot assembly in which structures are assembled by taking advantage of ambient environmental motions. Stochastic modular robotic assembly has been previously demonstrated in two dimensions on an air table (White, Kopanski, & Lipson, 2004; Klavins E. , 2007; Griffith, Goldwater, & Jacobson, 2005). Gilpin, Kotay, Rus, & Vasilescu (2008) followed a related approach that began with an ordered lattice and employed stochastic environmental motions to remove unwanted modules. We have previously demonstrated 3-D stochastic assembly in a fluid environment(Zykov & Lipson, 2007; White, Zykov, Bongard, & Lipson, 2005). However, the large scales of these systems (8-13 cm), and their reliance on oil as an assembly fluid (due to exposed electronics), led to intractably slow assembly rates and made it difficult to demonstrate the assembly of more than a few components.

In order to scale to larger numbers of modules while reducing their size, it is possible to further simplify the modules by removing all components that are active during assembly. However, this simplification comes at the cost of a more complex assembly substrate capable of module manipulation. In Part I of this dissertation, we described the use of this approach to demonstrate the 2-D assembly of 10-module

structures from 500 μm × 500 μm × 30 μm tiles on a microfluidic chip (Tolley, Krishnan, Erickson, & Lipson, 2008). These modules were self-aligning and attachment forces were provided by a passive latching mechanism on each tile edge. We described further experiments that explored the possibility of printing the microtiles with electronic components (Tolley, Baisch, Krishnan, Erickson, & Lipson, 2008), and the employment of hierarchical assembly for reducing assembly times and the cost of assembly errors.

In this chapter, we present an experimental system for expanding this scalable fluidic assembly approach to 3-D assembly. This system is composed of 1.5 cm scaled cubic modules which are assembled within a 1.3 L assembly tank (Figure 4.1). First we describe the module, assembly tank, and control software design for three iterations of the experimental system and the experimental testing of a number of the system's functionalities. The three major iterations are described in order to elaborate the reasons for the design choices made in developing the final system. Additionally, since the experiments presented later in this chapter were performed on various version of the experimental system, descriptions of these versions serve to further understand the experimental results.

The second part of this chapter describes five sets of experiments that use the experimental system to demonstrate a range of assembly operations in 3-D using the stochastic fluidic assembly concept. The first set demonstrates the capability of the system to attract a module to the system's assembly substrate and manipulate it reliably. Building on this, the second set of experiments demonstrates the capability of the system to assemble two- and three- module structures by inducing the modules to latch together at the assembly substrate. The third set demonstrates the capability of the system to repair a damaged structure placed in the assembly tank. For the fourth set of experiments, an upgraded version of the assembly system is used to assemble a

target non-planar structure. The reliability of this process is then assessed by analyzing the statistics for 10 consecutive assembly experiments. Finally, based on the discussion of the importance of hierarchical assembly in Chapter 3, the fifth set of experiments demonstrates the parallel hierarchical stochastic fluidic assembly of planar and non-planar structures.

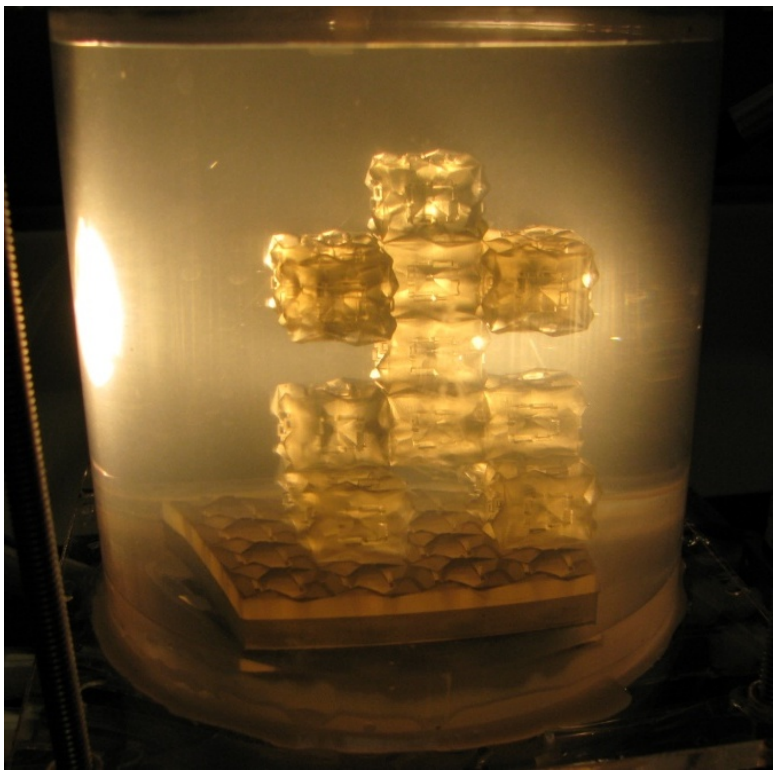


Figure 4.1. 3-D stochastic fluidic assembly system. Modules are assembled on an active patterned substrate on the bottom of a fluidic tank. Stochastic fluid motion is employed for module transportation. Fluid flow in and out of the substrate is modulated by a set of computer-controlled valves to direct structure assembly.

Module Design

The stochastic fluidic assembly modules were designed to be as simple as possible in order to accommodate scaling to large numbers and small dimensions. Thus, there

are no module components that must be active during assembly. In fact, we use single-material modules here in order to focus on the fluidic manipulation and assembly of target structures. Since the assembly process relies only on the module shape, we make the assumption that the modules can later be embedded with a variety of components necessary for their role within the robotic system (sensors, actuators, etc.).

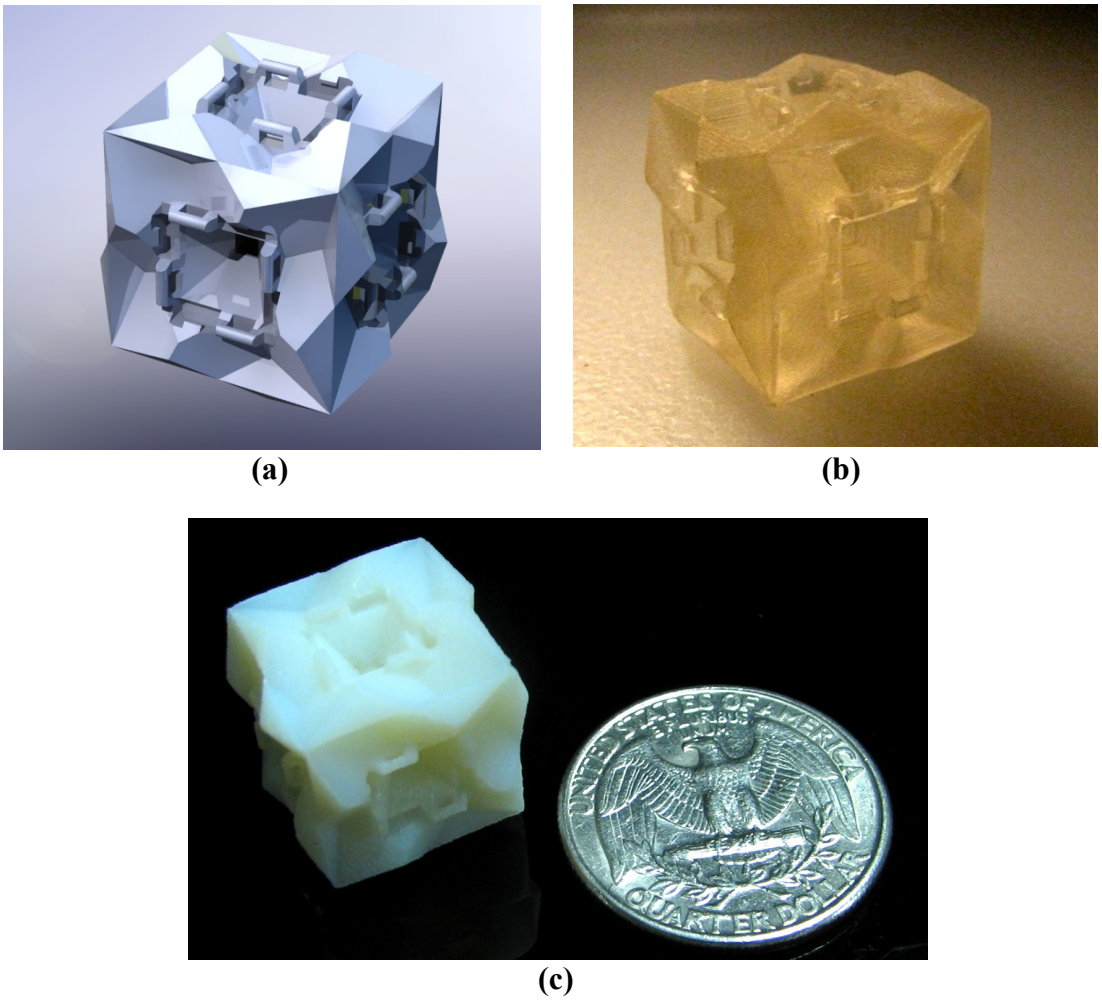
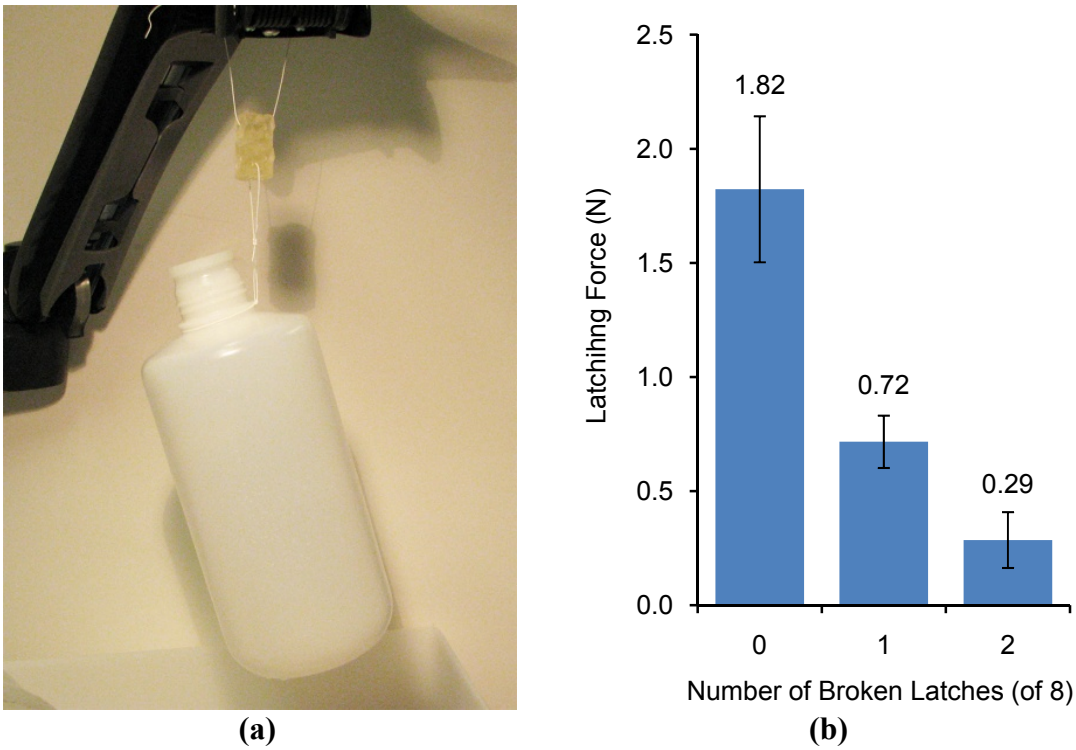


Figure 4.2. Module design. Images of (a) computer aided design, and (b) prototype modules 3-D printed out of Objet's FullCure 720, and (c) VeraWhite materials. Each has a side length of 15 mm. A set of extruded features on each face fit into complimentary cuts on adjacent modules in any orientation to promote alignment on a rectilinear lattice. Four latch loops on each face lock into complimentary protrusions to hold latched cubes together.

Latch Testing

We evaluated the latch design by measuring the total latching force between two modules. For these measurements we pressed the faces of two modules together manually, and suspended a weight from one cube while holding the other cube in the air [Figure 4.3(a)]. We increased the weight (by adding water to the suspended bottle) until the two modules became detached. Over 10 randomly selected face pairings, the latching mechanism provided an average latching force of 1.8 N, and a maximum force of 3.4 N.



(a) **(b)**

Figure 4.3. Latching force measurement. (a) The strength of the latches was measured experimentally by increasing the amount of weight suspended from a latched cube until the bond broke. (b) Plot of average maximum latching force with standard error indicated. The redundant latch design led to graceful degradation of the latching force with one or two broken latches.

Due to the mechanism's redundant design (i.e. with four independent latches per cube face), we expected its performance to degrade gracefully with damage. We tested this by measuring the force between modules with broken latches [Figure 4.3(b)]. We indeed found that there was still a force between the two cubes, although the average latching force was reduced to 0.7 N with one broken latch, and 0.3 with two.

Alternative Designs

Due to problems with broken latches, alternative latching mechanisms were considered. Figure 4.4(a) and (b) shows two designs based on the concept of a multitude mushroom-head latches that interlock like Velcro™. The advantage of this approach would be increased robustness to latch failure. However, the increased surface area required for latches led to cubes with very little geometry to enforce alignment [Figure 4.4(a)]. An additional problem inherent with this design was the possibility of latching in misaligned orientations. This led to the idea of a spherical module [Figure 4.4(b)], however the unconstrained relative module locations possible with this design would have overly complicated the assembly procedures.

A problem discovered in testing all of the designs was a large number of broken latches. However, the latches were breaking primarily on particular faces. Optical micrographs illuminated the cause of these failures [Figure 4.4(c)–(f)]. The 3-D printed material was found to have lower tensile strength between layers than within them. Thus, latches that were printed horizontally [Figure 4.4(c),(e)] were structurally sound, unlike many of those that were printed vertically [Figure 4.4(d),(f)]. Following this discovery, cubes with the original design (Figure 4.2) were printed at an angle such that none of the latches were printed vertically. Thereafter, very few latches were found to break during normal use, eliminating the need for an alternative design.

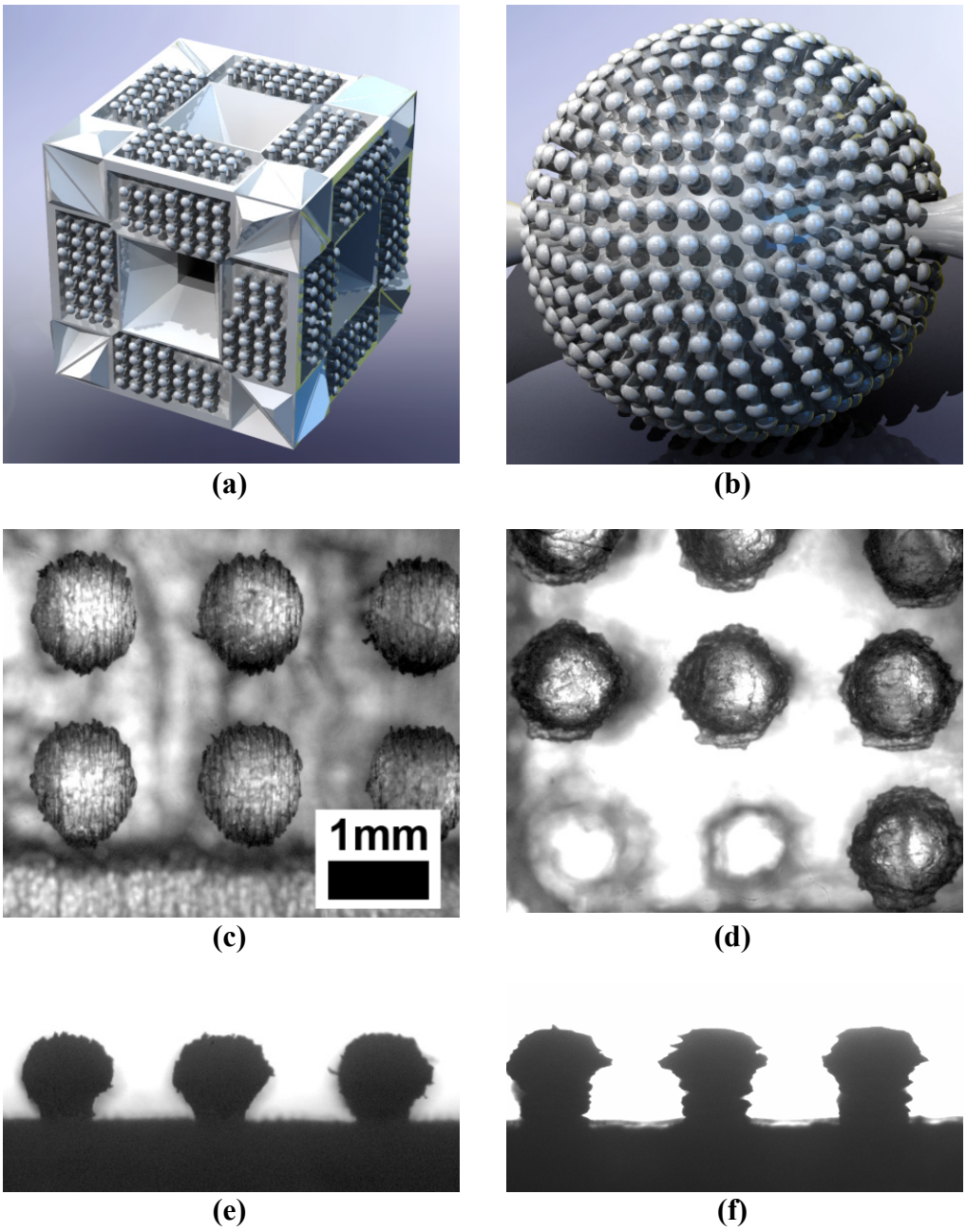


Figure 4.4. Alternative modules with mushroom-head latchs. (a), (b) Two alternative module designs were considered to overcome the problem of frequent latch failures. They both employ fields of non-gendered mushroom-shaped latches that interlock when pushed together. (c)–(f) Optical micrographs reveal the cause of frequent broken latches on particular faces: a lower tensile strength between layers than within them. Thus, latches printed vertically [(d) and (f)] were much less structurally sound than those printed horizontally [(c) and (e)].

Fluidic Assembly System – Version 1

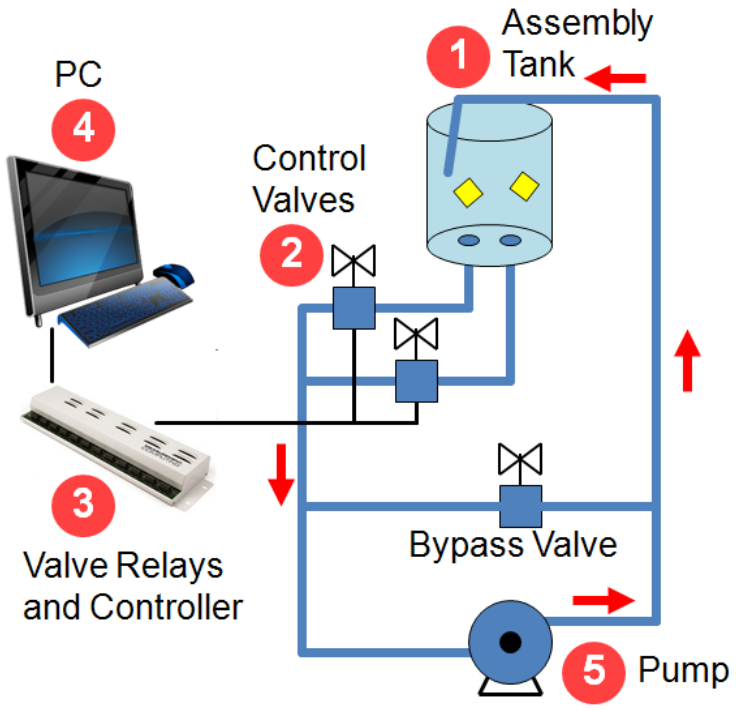
Our stochastic fluidic system assembles the modules described in the previous section by manipulating the fluid flow in a 1.3 L assembly tank. The experimental apparatus that accomplishes this is composed primarily of five parts: the assembly tank, a pump, a set of solenoid valves for flow control, relays and controller board, and a controlling PC (Figure 4.5). The pump is used to circulate the assembly fluid through the experimental apparatus. (We use tap water as an assembly fluid, see the *Assembly Fluid* section below for details.) The valves (controlled by the PC via the relay controller) are used to switch the flow path through the assembly tank in order to indirectly manipulate the modules. Table 4.1 summarizes the off-the-shelf components used in all of the versions of the fluidic assembly system.

Module assembly occurs on a patterned substrate at the bottom of the assembly tank. The substrate consists of a four by four array of ports patterned to match the cube faces, but without the latching mechanism (Figure 4.6). A pyramid at the centre of each port inserts into an indent in the centre of each cube face to further improve alignment. Each port also has four fluid channels (one at each corner) that connect to a single channel on the outside of the tank. A valve (normally closed) connected to each port controls the flow of fluid through the port. A single source at the top of the tank, connected to the pump outlet, provides stochastic circulation in the tank that transports the modules.

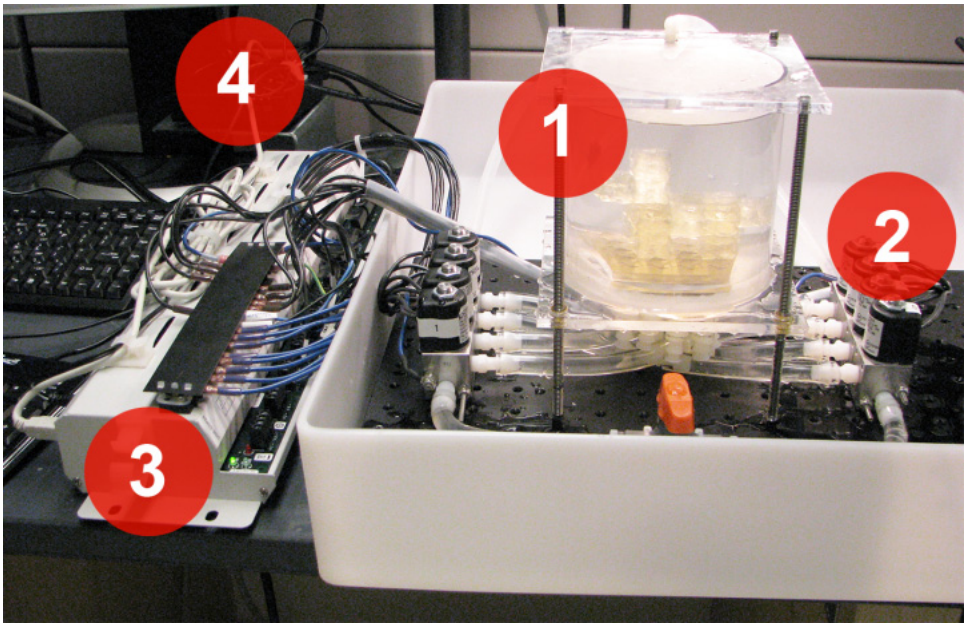
The tank itself is not sealed such that it is kept at atmospheric pressure. This facilitates the introduction and removal of modules and assemblies into, or out of, the tank. A hand-operated valve on a separate flow line connecting the high and low pressure ends of the pump allows for the regulation of the amount of flow through the tank.

Table 4.1 Off-the-Shelf System Components

Component	Version Added	Manufacturer	Part	Relevant Quantity	Value
Gear Pump	1	Oberdorfer Pumps Inc.	N991-F41	Max Flow Rate (gpm)	2
Pump Motor	1	Baldor Electric Co.	17E537W46 0G1	Power (W)	249
Solenoid Valves	1	Peter Paul Electronics Co. Inc.	52X05570G B	Flow Coefficient (Cv)	0.292
Solid State Relays	1	Measurement Computing Co.	SSR-OAC-05	Max Switching Time (ms)	8.33
USB Controller Board	1	Measurement Computing Co.	USB-SSR24	Relay Module Capacity	24
PC	1	CompuLab	Fit-PC2	Processor Speed (GHz)	1.6
Pressure Sensors	2	Omega	PX40-50BHG5V	Pressure Range (mmHG)	+/- 50
Analog Input Module	2	Measurement Computing Co.	USB-1208FS	Sampling Rate (kHz)	50
Air-bubble-release Valve	3	Honeywell	PV 075	Flow Coefficient (Cv)	13
Air-Powered Diaphragm Pump	3	Ingersoll-Rand	PD02P-HPS-PTA	Max Flow Rate @ 50 psi (gpm)	2.7

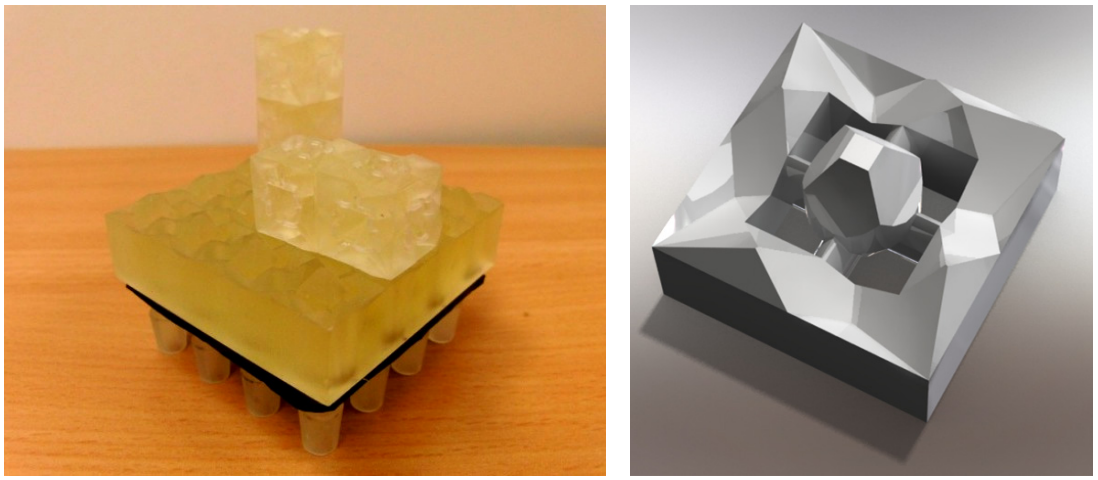


(a)



(b)

Figure 4.5. Experimental Apparatus. (a) Schematic and (b) photograph of experimental apparatus consisting of 1) an assembly tank, 2) control valves, 3) valve relays and USB controller board, 4) PC and keypad, and 5) fluid pump (not pictured in photograph).



(a) (b)

Figure 4.6. Assembly substrate. (a) 3-D printed substrate component with 16 ports shaped to complement cube faces. (b) Rendered CAD drawing of one of the outlet ports with a centre pyramid for alignment and four corner channels.

Control software

A custom valve control program was written in C++ to switch the states of the valves independently using a keypad or mouse and give feedback on their states (Figure 4.7). When a valve is opened, the corresponding button's colour switches from red to green. An option to switch chosen valves on and off continuously at a specified frequency was also included. This program calls the library functions provided by the controller manufacturer to energize the appropriate relays to switch the desired valves on or off.

Testing

With the regulation valve closed, we would expect the flow rate at each port acting as a sink to decrease linearly with the total number of sinks open as the number of possible fluid paths increases. We tested this experimentally by recording the time taken to pump one litre of water through the substrate with one to four valves open (Figure 4.8). Our results indicate that this is indeed the case.

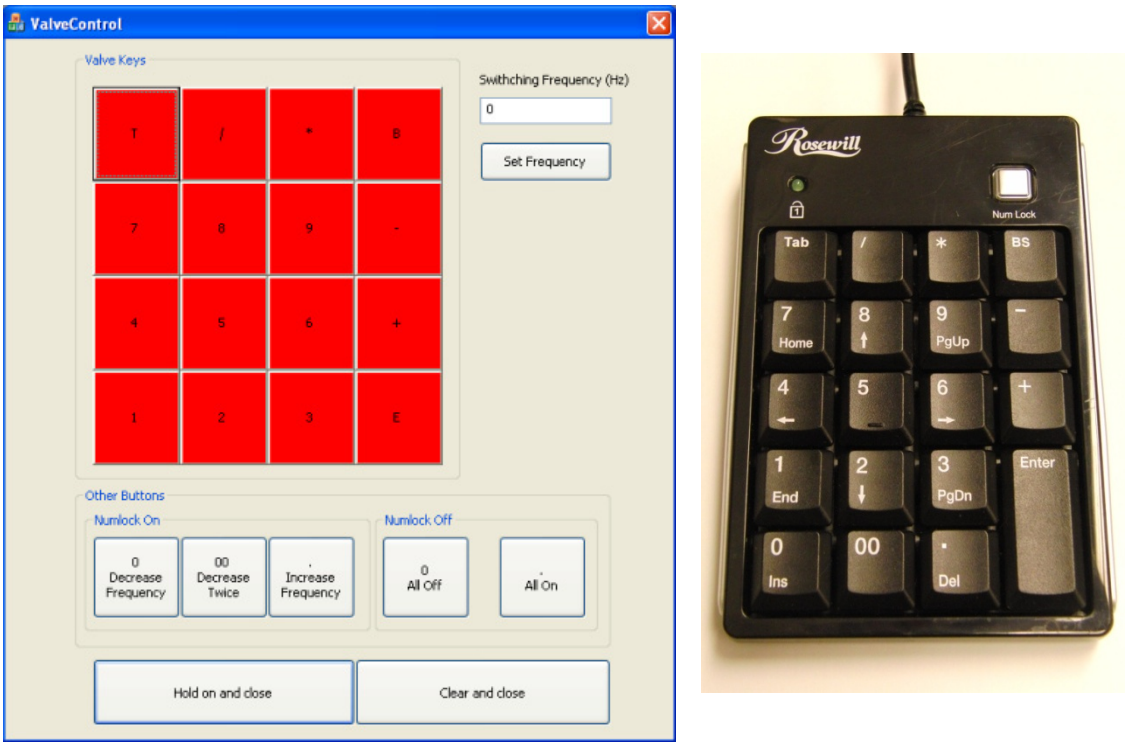


Figure 4.7. Version 1 custom valve control software. A custom valve control program (left) communicates with the USB valve relay controller to switch the valves in response to user commands from a keypad with a similar physical layout (right). Additional buttons are used to set the valve switching frequency.

Experiments with the initial version of our experimental system demonstrated it's capability for module manipulation and simple assembly tasks (see *Manipulation Experiments* section below for details). However, some major limitations were identified. First, the motions of the modules were very sensitive to the position of the inlet tube at the top of the tank. For certain inlet tube positions, the cubes would become stuck in locations with little fluid circulation. Second, a lack of sensor feedback meant that humans would be required in the loop to control all assembly operations. Third, the routing of fluid from the substrate ports to the control valves was complicated and difficult to assemble or disassemble.

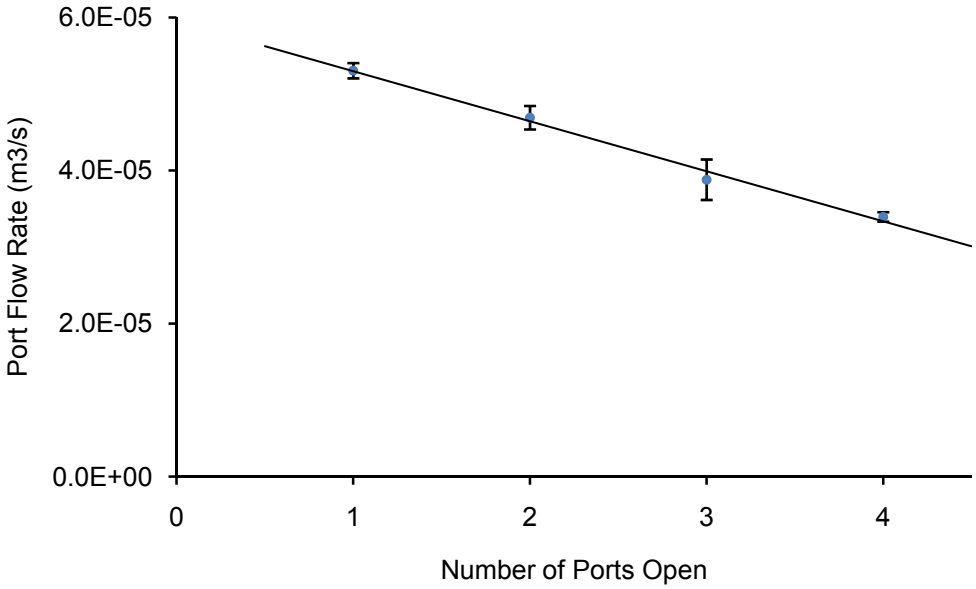


Figure 4.8. Sink flow rate vs. number of ports open. As expected, we found the flow rate through each port acting as a sink to decrease with the number of ports open.

Fluidic Assembly System – Version 2

In order to address some of the shortcomings of our initial system design, we developed a second experimental system we refer to here as *Version 2*. The primary components of the upgraded system, as labelled in Figure 4.9, are 1) the assembly tank, 2) solenoid valves to control fluid flow into and out of each substrate port, 3) a valve relay control board, 4) a PC for executing assembly control algorithms, 5) miniature voltage-output pressure sensors at each assembly port, 6) an analog input module, and 7) a circulation pump. Comparing Figure 4.9(b) with Figure 4.5(b) we see many small improvements in the layout of the system, the separation of the electronics from the fluidics, etc. Here we discuss the primary upgrades and how they addressed the problems identified in the previous section with the initial version.

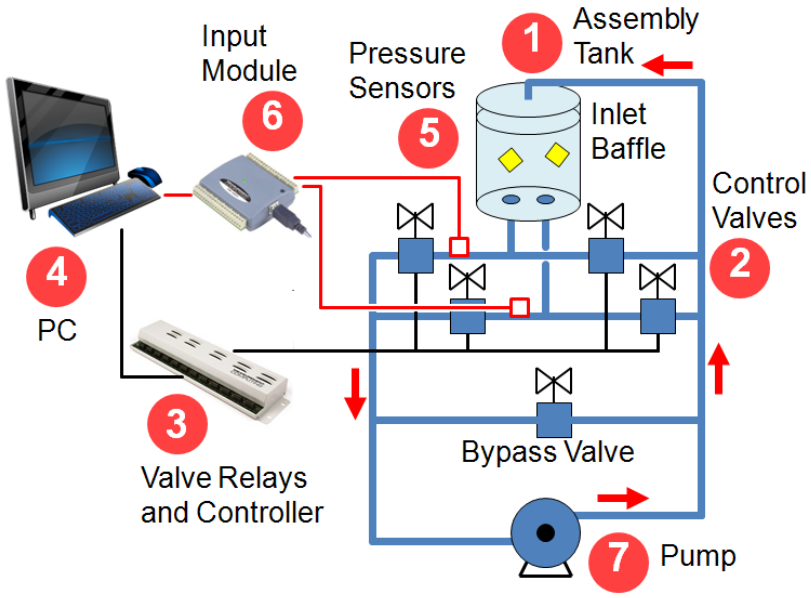
In order to deal with the problem of cubes getting stuck during assembly, we changed the system connections such that each substrate port could act as either and

inlet or an outlet. Thus the substrate itself could be used to agitate the flow. This was achieved using two valves (normally closed) associated with each port that could be opened to connect the port to either the high or low pressure end of the pump [see Figure 4.9(a)]. This allowed each port to act as a source, a sink, neutral, or be deactivated, depending on the states of the associated valves. 18 valves were needed for a 3 × 3 port substrate. We added another six valves (increasing the total up to the relay controller's maximum of 24) in order to control various other lines for fluid inlets, testing, and maintenance.

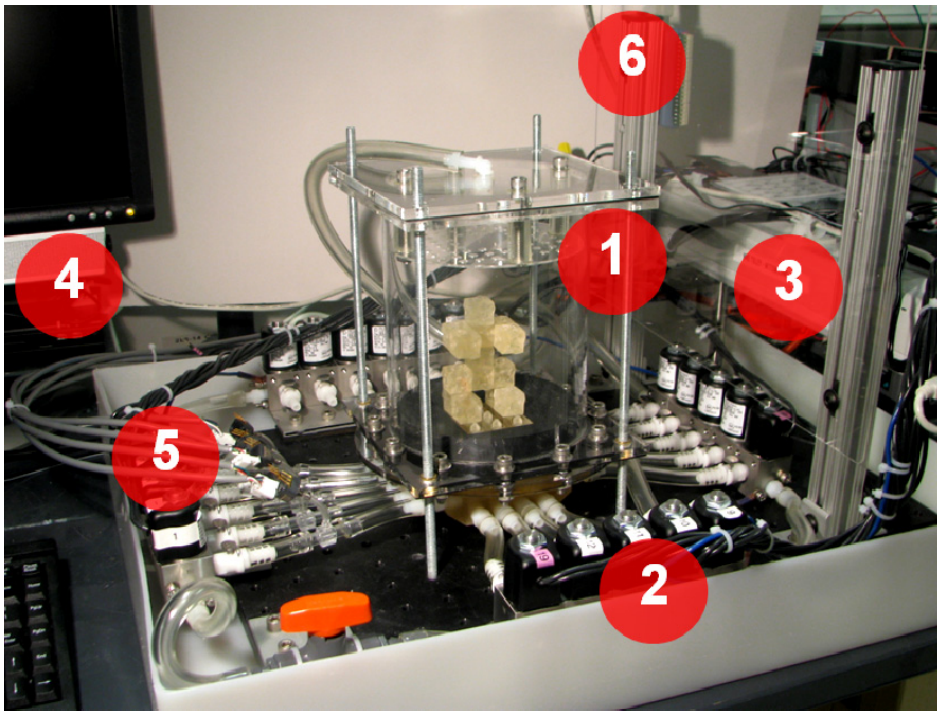
Additionally, because we no longer relied on the inlet flow for agitation, we were able to remove the inlet tube and add a baffle to the top of the tank to distribute the inlet flow evenly and eliminate the system's sensitivity to the inlet position. This baffle had the added effect of preventing many bubbles from entering the main portion of the assembly tank.

The second major upgrade was a complete redesign of the 3-D printed substrate. In addition to optimizations of the port openings on the substrate to maximize flow rates, the 3-D printed structure was enlarged to internalize the complex routing to the valves (Figure 4.10). Female NPT threads connections were integrated into the substrate to facilitate connection with the solenoid valves. This upgrade was necessary to make the increase from one to two valves per substrate port tractable.

The third major upgrade was the addition of pressure sensors to the tubes connected to the substrate ports. Experiments have shown that a single pressure measurement at each substrate port is sufficient to identify the presence or absence of a cube which may be sufficient for many assembly operations. For details, please see the *Feedback Sensing* section below.

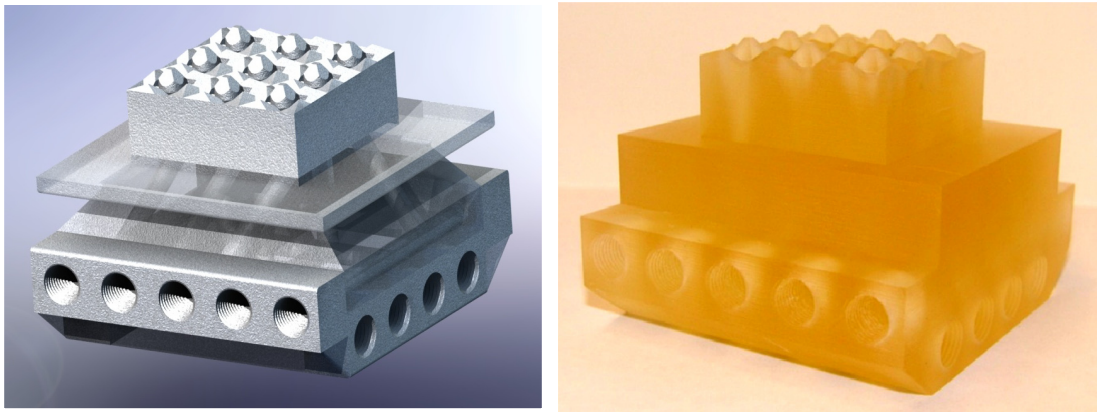


(a)



(b)

Figure 4.9. Version 2 of experimental setup for fluidic assembly of minimalistic components. (a) Schematic and (b) photograph of upgraded system consisting of 1) an assembly tank with inlet baffle, 2) solenoid valves, 3) valve relay control board, 4) PC for valve control, 5) voltage-output pressure sensors, 6) analog input module, and 7) Circulation pump (not visible in photograph).



(a) **(b)**

(c)

Figure 4.10. Version 2 substrate. (a) CAD model and (b) 3-D printed version of upgraded assembly substrate at the bottom of the assembly tank. (c) Close-up image of substrate with black 3-D printed funnel to prevent cubes from becoming stuck at the bottom of the tank.

Control Software

As mentioned in the previous section, the number of valves in Version 2 increased from 16 to 24. Thus, our manual valve control software and GUI was also updated to reflect this change (Figure 4.11). The Version 2 software allows the operator to switch the state of each substrate port to be an inlet, an outlet, neutral, or closed. Six additional valves are used to control the global inlets and outlets at the top of the tank as well as maintenance lines. All valves can be controlled with either mouse or

keyboard input and can have their button mappings changed on the fly. As in the first version of the software, additional controls allow the valves to be switched on and off at a specified frequency to help snap modules into place.

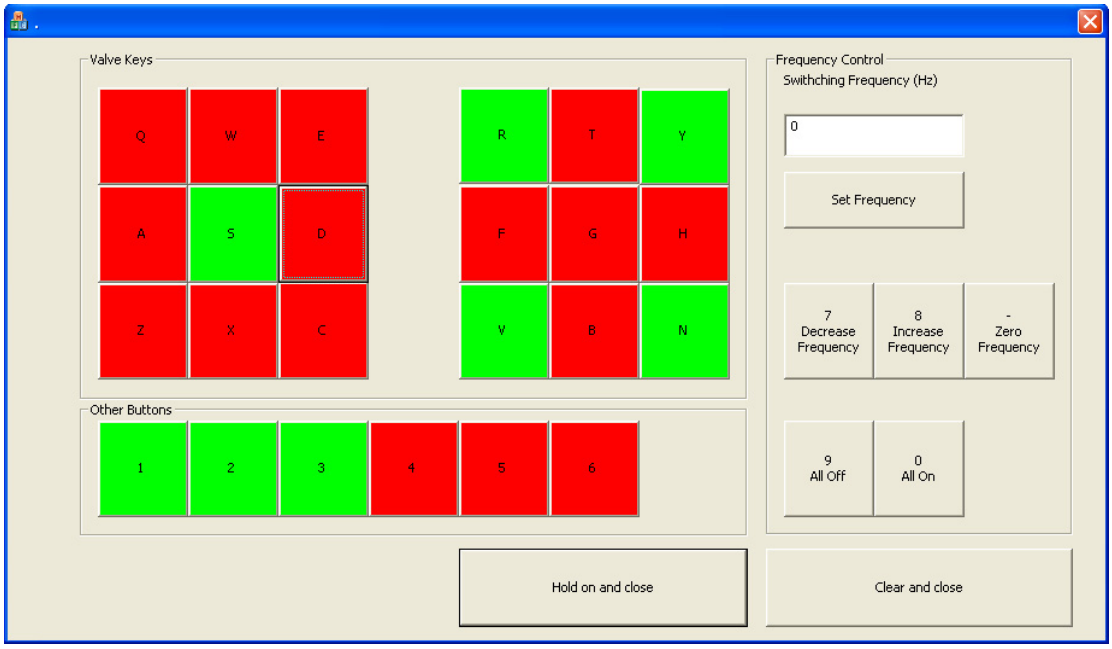


Figure 4.11. Updated valve control software for manual actuation of the 24-valve Version 2 system. The nine coloured buttons on the top left map to the nine outlet ports opening or closing them to the inlet flow. Similarly, the nine coloured buttons on the top right open the valves to the outlet flow. The other coloured buttons open or close the miscellaneous valves while the remaining buttons control valve frequency switching.

Testing

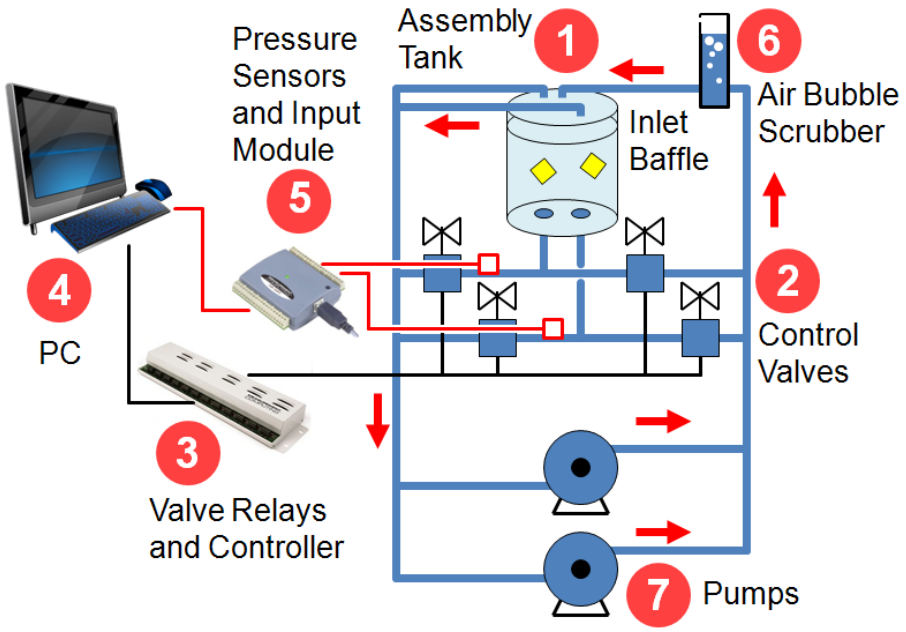
As before, experimental testing identified problems with the Version 2 of our experimental system. Primary among these were insufficient fluid flow for certain assembly operations, poor experiment imaging, inadequate space on the substrate, and imprecise control of valve switching. The next section on the third version of our system describes these problems in more detail along with the upgrades made to address them.

Fluidic Assembly System – Version 3

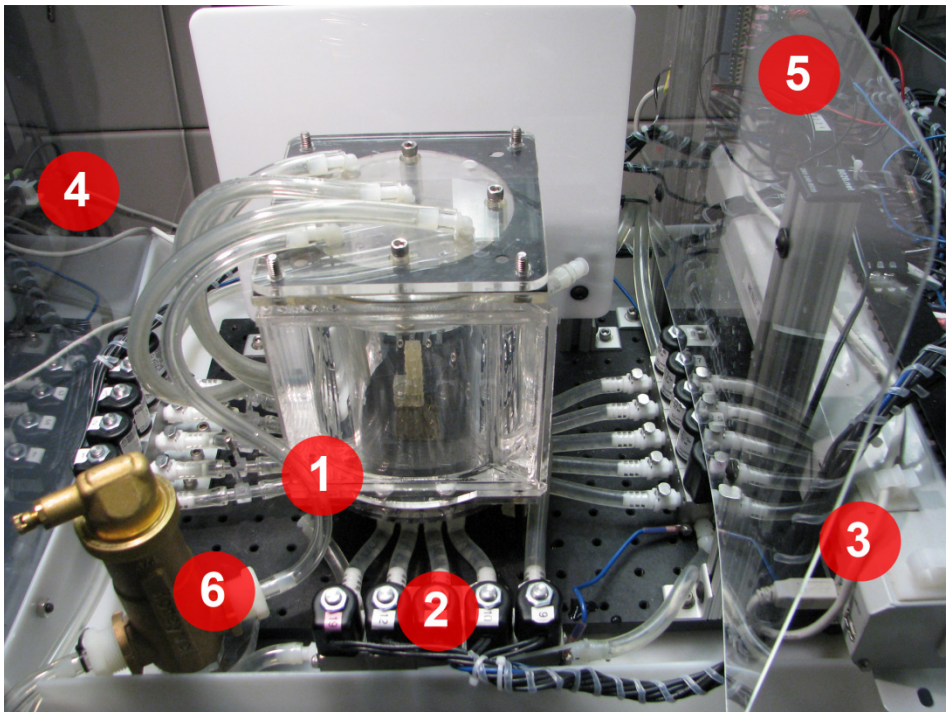
Based on extensive testing with the second version of the experimental systems, a number of significant changes were made to the structure and components of our fluidic assembly system that we refer to here as Version 3. Figure 4.12 shows a schematic and photograph of the upgraded system. In addition to small cosmetic and practical improvements to the system (e.g. more splash shields), six major changes were made.

The first two changes were made to address the problem of insufficient flow through the system. During some experiments with many ports open simultaneously and during many latching operations, the fluid flow was insufficient to manipulate the modules. The first approach to solving this problem was to add two always-open fluid outlets at the top of the tank [as seen in Figure 4.12(b)]. This significantly increased the overall fluid flow which somewhat counter-intuitively increased the control of cubes at the substrate. As a second measure, we also added a second pump in parallel with the first. For the second pump, we chose an air-powered diaphragm pump. Unlike a gear pump, a diaphragm pump emits pulsating flow which can help to dislodge modules or latch them together. Also, by regulating the air pressure supplied to the pump, we are able to adjust its contribution to the overall flow, unlike with the gear pump which was powered by an electric motor at a constant potential.

Unfortunately, increasing the flow through the system has the side effect of adding bubbles that hinder attempts to observe or image experiments. In order to address this problem we installed an ultra-efficient air vent with a micro-bubble concentrator manufactured by Honeywell. This part traps small bubbles on a high-surface-area internal structure, accumulates them, and releases them through a release valve. The vent had the added benefit of simplifying the removal of air from the water lines.



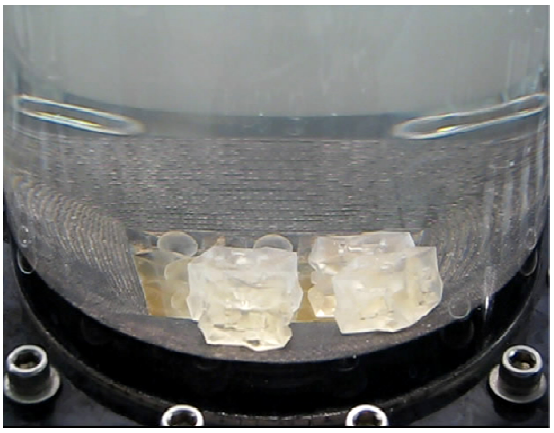
(a)



(b)

Figure 4.12. Version 3 of experimental system. (a) Schematic and (b) photograph of updated experimental system with main components identified. The primary upgrades with Version 3 are the addition of a second pump, a bubble scrubber, and a water lens for imaging, as well as a re-routing of the fluidic circulation.

(a)



(b)

(c)

Figure 4.13. Water lens. (a) A custom water lens was fabricated to reverse the optical distortion of experiment videos due to the cylindrical assembly tank. (b) Before, and (c) after images demonstrate the effectiveness of the water lens.

Another problem with experiment imaging was the visual distortion caused by the cylindrical shape of the experimental chamber. In order to reverse the optical distortion, we designed and fabricated a custom water lens for our experimental

assembly tank (Figure 4.13). This was accomplished by creating a secondary fluid tank out of acrylic with a cylindrical interior and a planar exterior such that the light rays bent at the water-acrylic-air interface were straightened at the air-acrylic-water interface. However, the side effect is small aberrations at the tank sides [see Figure 4.13(c)].

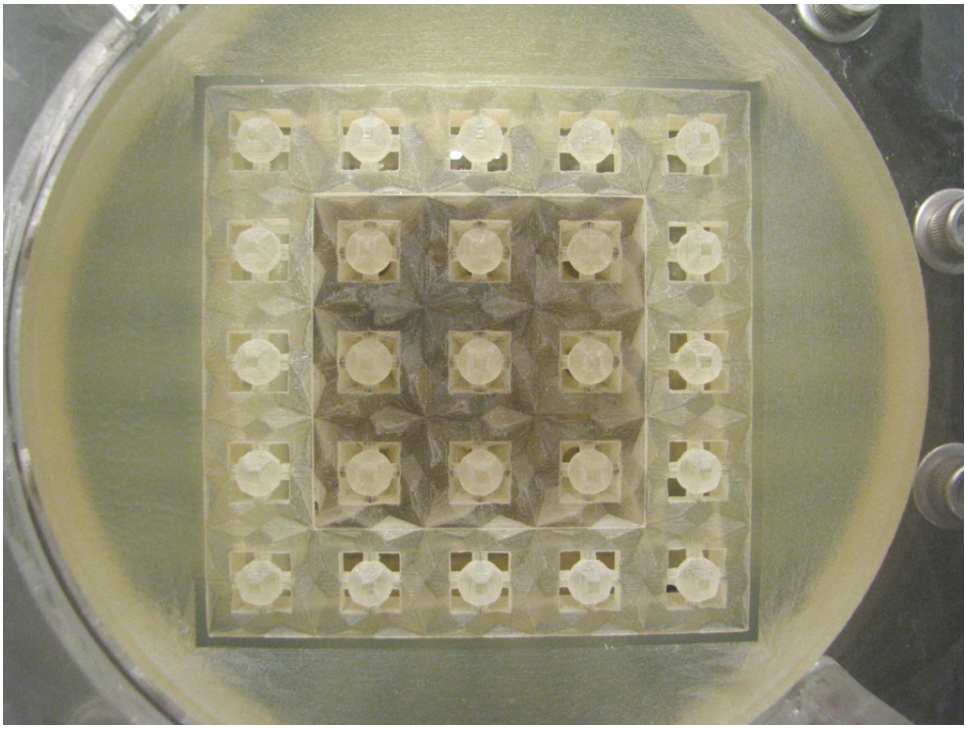


Figure 4.14. Version 3 extended substrate. A perimeter of passive ports around the active assembly substrate allow the manipulation of larger sub-assemblies.

In order to increase the range of experiments possible with Version 3 of our experimental system, we designed and printed a new version of the funnel that surrounds the assembly substrate (Figure 4.14). This new funnel extends the 3 × 3 ports of the assembly substrate with a surrounding perimeter of passive ports. This allows the assembly of structures larger than three cubes long, as long as they can be manipulated by the nine ports in the centre of the substrate.

Control Software

The last issue addressed in Version 3 was the problem of poor control resolution. Since valves were always either on or off, in many situations a module that needed a slight nudge into place would get a huge jet that would eject it completely from the substrate. In order to achieve a more continuous range of control options, we implemented pulse-width modulation (PWM) of the control signals to the valves. This was achieved by opening and closing the valves rapidly at a frequency that resulted in an approximately continuous flow of fluid. In the next section we discuss PWM tests in which we found that ON-OFF cycles frequencies of 5, 10, and 20 Hz all worked for this purpose. We then adjusted the amount of fluid flow by changing the duty cycle (i.e. the fraction of the switching cycle spent in the ON position).

The GUI for the Version 3 custom control software with PWM is shown in Figure 4.15. The sliders at the bottom-left allow the nine inlet valves to be set to 11 settings ranging from 0 (continuously OFF) to 1 (continuously ON). We also experimented with using PWM on the outlets but did not find this to be particularly helpful. One of the side-effects of the modulated flow is that the normally smooth flow (if using only the gear pump) becomes pulsated, which is useful for overcoming friction when manipulating modules.

The other main development with Version 3 of the valve control software was a set of open-loop algorithms for various stages of an assembly operation. The approach was to minimize the feedback required for assembly by devising a set of repeating valve sequence to achieve a particular assembly task such as attraction, alignment, latching, manipulation, etc., and switching between them to act as a sort of finite state machine (FSM). For example, for the attraction state, the user identifies the attraction locations and clicks the *start* button. Those locations then become sinks while the rest of the substrate becomes sources. In order to agitate the cubes, the sources cycle

between levels of low and high flow. Once cubes have been attracted to the desired locations (this can be identified either with human or sensor feedback), the software moves to the next stage of assembly (alignment). Thus the GUI shown in Figure 4.15 has a list box with a set of possible states on the bottom-right, along with a set of sliders for adjusting the overall parameters of each open-loop algorithm (e.g. the upper and lower limits on the amount of source agitation in the attraction case). See the *Robust 3-D Assembly* section below for further details on the use of this FSM approach for the assembly of 3-D structures.

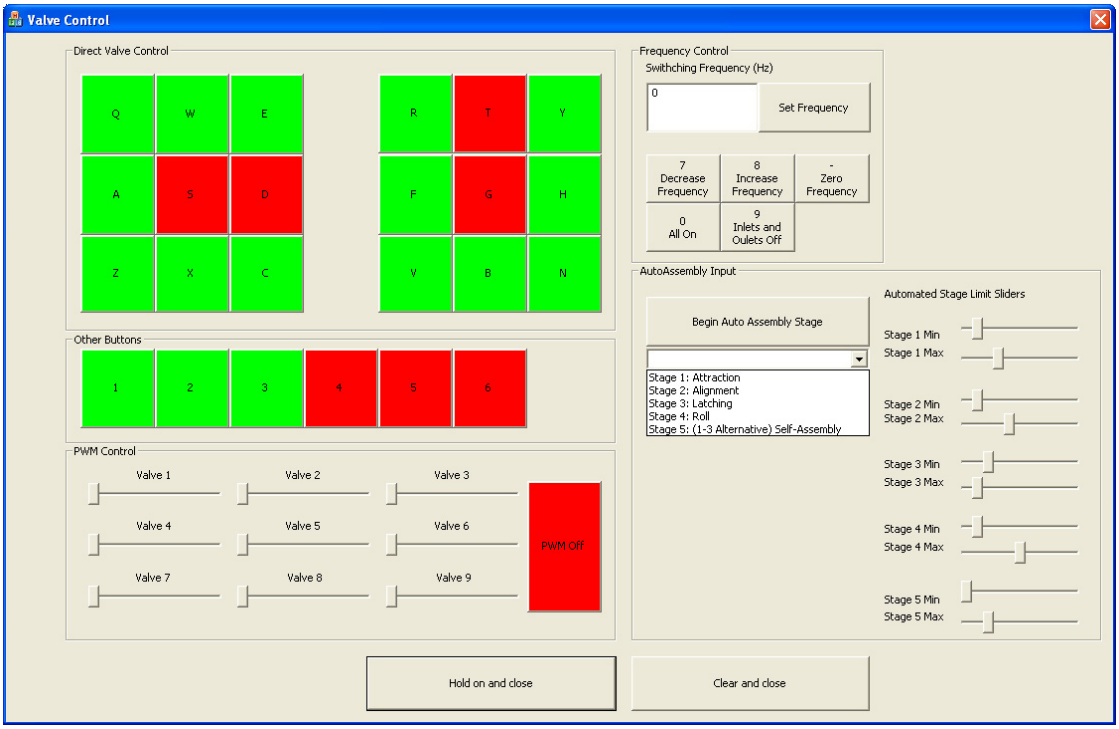


Figure 4.15. Version 3 Control Software GUI. Additional controls allow pulse-width modulation of inlet flows, as well as execution of preset open-loop assembly sequences.

Testing

We tested the PWM valve control by connecting a flow meter to the outlet of a

valve and measuring the resulting steady-state flow rates for various PWM settings. Figure 4.16 shows the results of these tests for PWM with three different frequencies (5 Hz, 10 Hz and 20 Hz). We found the flow rates to be monotonically increasing and fairly close to linear in the PWM fraction. As expected, the results showed the same trend for each of the three frequencies (indicating that any of these frequencies switch sufficiently fast for PWM).

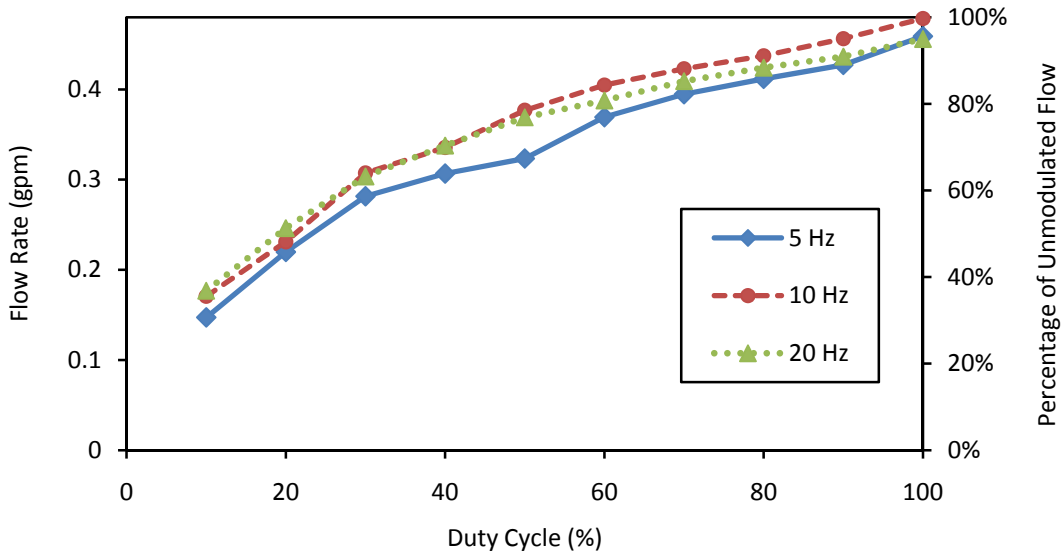


Figure 4.16. Testing of Pulse-Width Modulation (PWM) Control. Experiments showed PWM of valving was able to

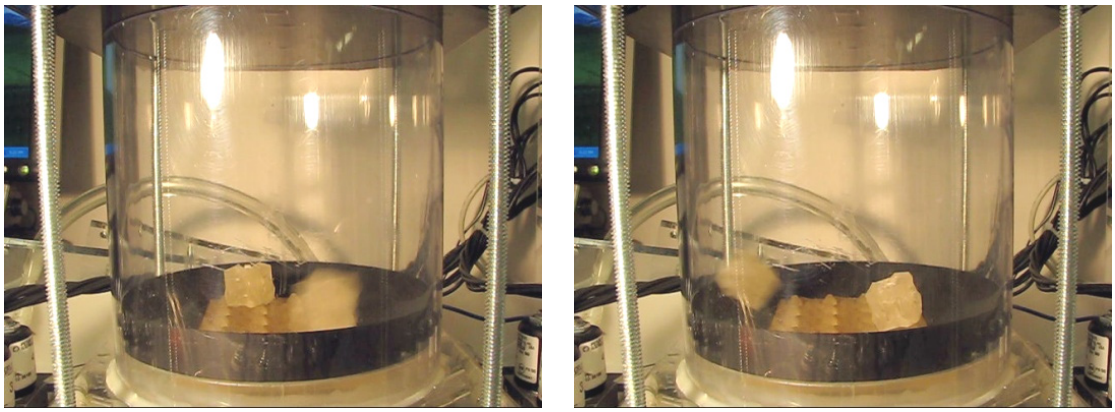
Interestingly, the tests all showed a significant amount of flow at 10% PWM fraction (i.e., closer to 35% of the average unmodulated flow rate). In fact, the data seems to match an exponential curve with a power of 0.5. Using this information, duty cycle values could be used that lead to a more linear modulation of the flow rate between the open and closed values. However, this would require the solenoid valves to switch up to an order of magnitude faster (e.g. after 0.001 s instead of 0.01 s for the 10 % level with PWM switching at 10 Hz). Since the switching rate limit of the solenoids was met in these experiments, this would require slowing the overall PWM

switching rate by a factor of ten, which would lead to less continuous outlet flow. In any case, experiments determined that this unmodified modulation relationship was sufficient for assembly (see the *Assembly Experiments* section below for details).

Assembly Fluid

Tap water was chosen as the assembly for various reasons. First, since this assembly approach was inspired by biological assembly, water was a natural choice. It is readily available, inexpensive, safe, and compatible with the widest range of off-the-shelf hardware components (e.g. pumps, valves, etc.). Nonetheless, we did consider some alternatives that have advantages with respect to water. Oil, for example, was used in previous stochastic assembly experiments (White et al. 2005, Zykov and Lipson, 2007) as a non-conductive alternative to water for compatibility with exposed electronics. However, since the experimental system discussed in this chapter does not require active components during assembly, the complications that necessitated working with oil were not present. Additionally, the increased viscosity of oil in previous work (including the microscale experiments presented in Chapter 1), slowed down the movement of the modules and consequently assembly rates.

Another assembly fluid we considered was air. The advantages of using air as an assembly fluid include increased portability, reduced infrastructure requirements, and increased compatibility with electronics and corrodible materials. The ability to use air instead of water would lead to a simpler, more versatile system. Thus, we conducted preliminary experiments to investigate this possibility (Figure 4.17). Our initial results indicated that the increased fluid flow required to move our modules with air led to high-energy stochastic module motions which were difficult to control. We found it very difficult to attract modules to a particular location and we were not able to make them latch together. For these reasons, we decided to upon water as an assembly fluid.



(a) **(b)**

Figure 4.17. Air experiments. Assembly experiments were conducted to investigate the possibility of using air as an assembly fluid.

Feedback Sensing

In order to implement assembly with feedback control, the system needs to be able to detect when a module has been attracted to the substrate. To detect cubes, we added single-ended voltage-output pressure sensors to the outlets of each substrate port. When a cube is attracted to a substrate port, it restricts the flow, thus affecting the pressure in the outlet line. These changes are detected by the sensors and sent to the controlling PC through an A/D input module.

We tested the ability of the embedded pressure sensors to detect the state of the system by recording their outputs for various configurations (Figure 4.18). For these experiments, we ran the system with all substrate ports acting as outlets and recorded the average sensor signals at five ports (the centre and four adjacent). We repeated this process five separate times for each configuration and plotted the average values. For this proof of concept we took the empty tank [Figure 4.18(a)] measurements as a baseline and compared the other measurements against those. Thus the measurements represent differential pressure readings with respect to the baseline.

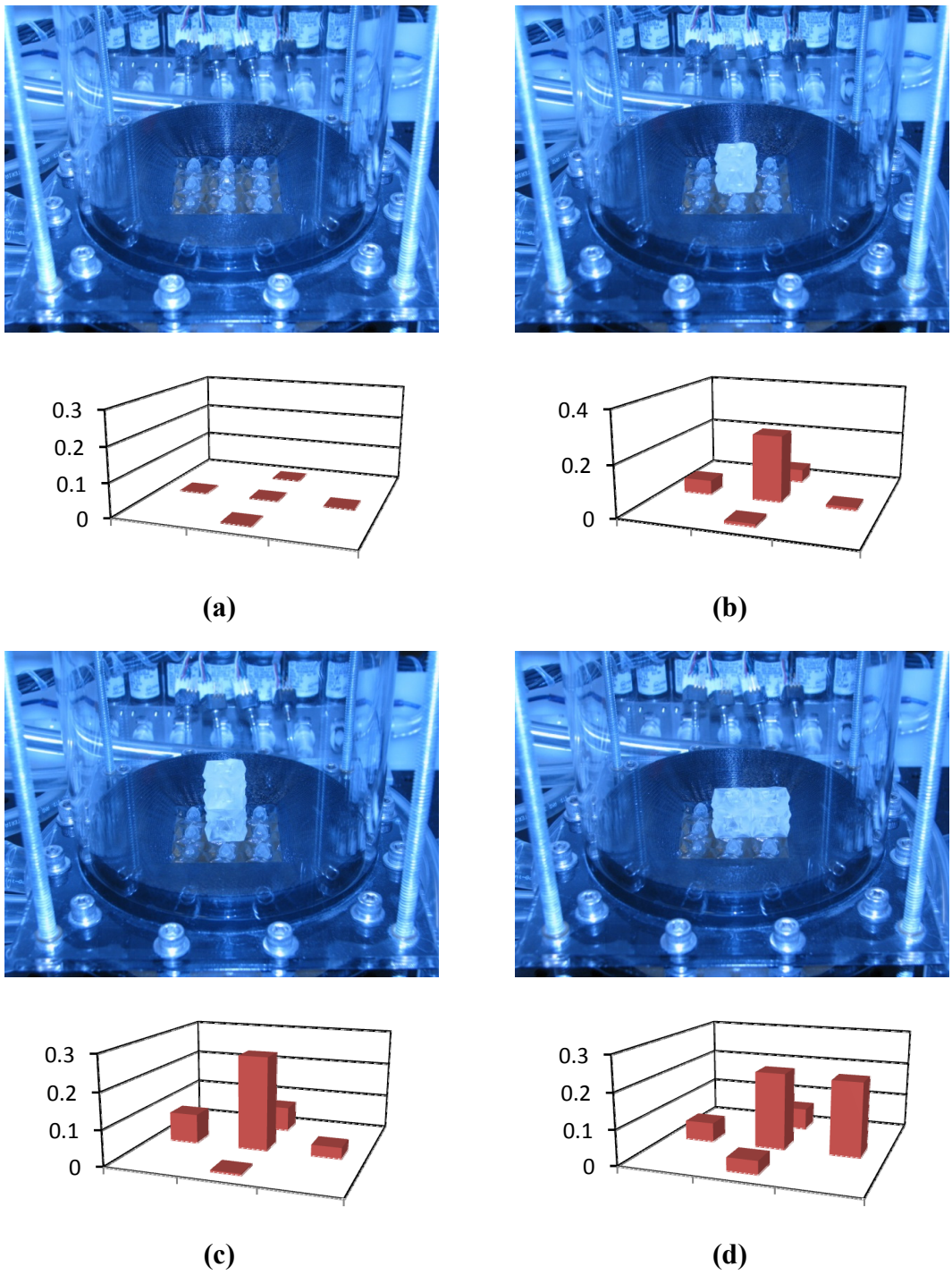


Figure 4.18. Automatic assembly configuration detection. Pressure sensors at the substrate fluid outlets were used to detect the presence of assemblies. Photographs (without water in the tank) illustrate the four test configurations: (a) baseline, (b) single cube, (c) vertical pair, (d) horizontal pair. The plot below each photograph shows the corresponding differential sensor measurements.

We see in Figure 4.18 that the pressure sensors are clearly able to distinguish between the presence and absence of a cube at the substrate. Additionally, it is interesting to note that there is a small but seemingly significant difference between the measurements of the configurations shown in subfigures (b) and (c). This indicates that it may be possible to obtain information about the configuration of the tank in three dimensions using only measurements at the substrate. However, this conjecture remains to be confirmed by further experiments.

Manipulation Experiments

The first set of experiments was conducted to evaluate the ability of the fluidic assembly system to reliably manipulate individual modules, in spite of the stochastic agitation (Figure 4.19). These experiments were performed with Version 1 of the experimental system described above. For each experiment, a single module was introduced into the tank, and a substrate port was turned into a sink. Once the module was attracted to the substrate, the sink was closed, and simultaneously an adjacent port was opened as a sink to attract and align the module. The same procedure was then repeated with adjacent ports to "juggle" the module among the four ports in the centre of the active substrate until it became stuck (did not move to the next position). The module was then released from the substrate by closing the outlets and allowing the stochastic agitation to carry the module away. The total number of successful port moves was recorded, as was the fraction of these moves that alignment patterns caused the cube to mate spontaneously with the substrate.

Results and Discussion

We found that our system was able to manipulate modules quite effectively in the stochastic environment, and that the modules frequently aligned spontaneously with

the substrate (Table 4.2). Over nine experiments, switching the valves as described above moved the modules an average of 19 times in the intended direction before they became stuck and had to be ejected off the substrate. Furthermore, the modules aligned spontaneously with the activated substrate port an average of 72% of the moves in each experiment, without the use of vibration or flow pulsation.

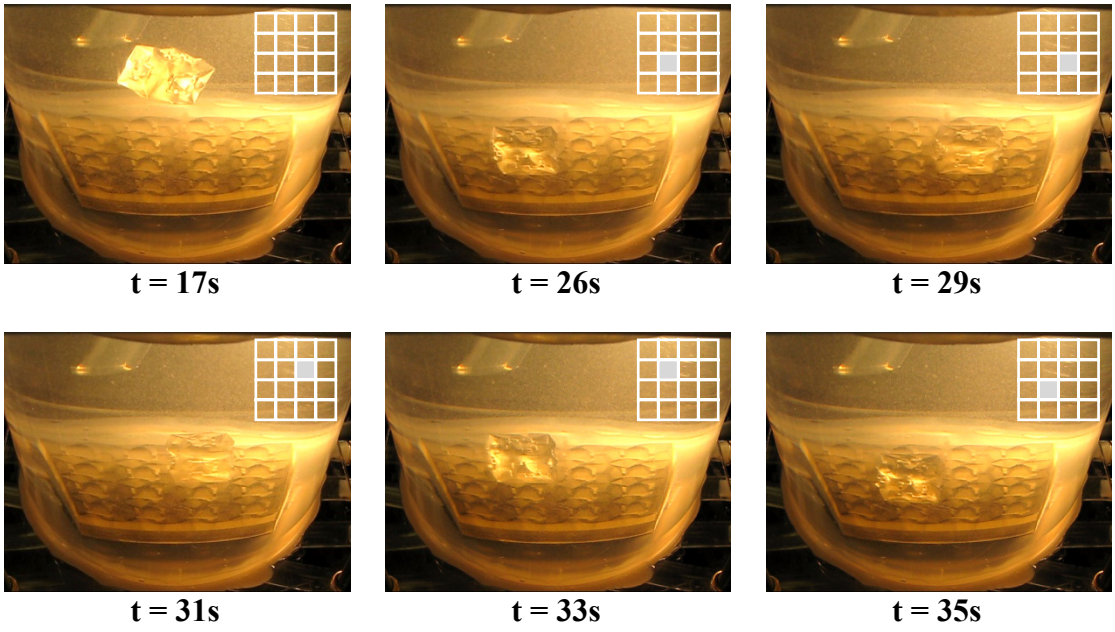


Figure 4.19. Module manipulation. Sequence of video frames from an experiment demonstrating the manipulation of a module by circulating it among four substrate ports. Filled squares in grid overlay indicate substrate locations occupied by the module.

Table 4.2 Manipulation Experiment Results

Measure	Mean Value	Standard Error
Number of Successful Moves	19.2	6.1
Alignment Rate	71.7 %	9.8 %

Assembly Experiments

The second set of experiments evaluated the ability of Version 1 of the assembly system to fabricate two- and three-module structures. Pairs of modules were first assembled by introducing four modules into the tank and opening two sinks (Figure 4.20). Once two modules were attracted to adjacent sinks, the control software's frequency option was used to repeatedly open and close the sinks, which had the effect of vibrating the cubes. The switching frequency was then adjusted from 0 to 40 Hz until the cubes were latched together. The ports were then closed in order to release the structure from the substrate into the agitated fluid environment. If assembled, the total latching times were recorded. L-shaped three-module structures were assembled in a similar manner, by adding a third cube to an assembled pair (Figure 4.21).

Results and Discussion

The assembly experiments demonstrated the ability of our stochastic system to assemble 3-D structures. By attracting cubes together on the substrate and switching the corresponding valves on and off at an appropriate frequency for a sufficient duration, it was generally possible to induce adjacent cubes to latch. Table 4.3 lists the average time required to assemble the two- and three-module structures.

Experimenting with various vibration frequencies to induce modules to latch led to an interesting observation: At lower frequencies, the modules have larger motions (thus can correct for larger misalignments) but are less likely to latch together if aligned (since they move more slowly and thus have less kinetic energy when they collide). By contrast, at higher vibration frequencies, aligned modules are more likely to latch but misaligned modules are less likely to align. Thus, increasing the vibration frequency has an effect reminiscent of annealing where if it is done at the correct rate, the modules tend to align nicely on the cubic lattice before latching together.

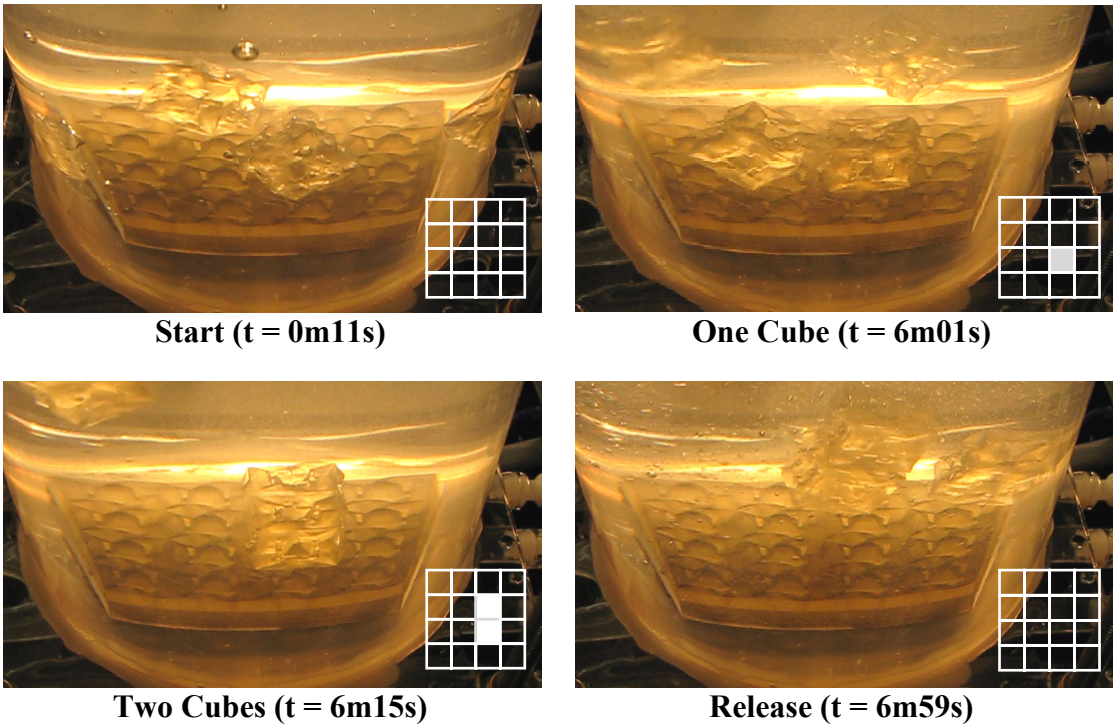
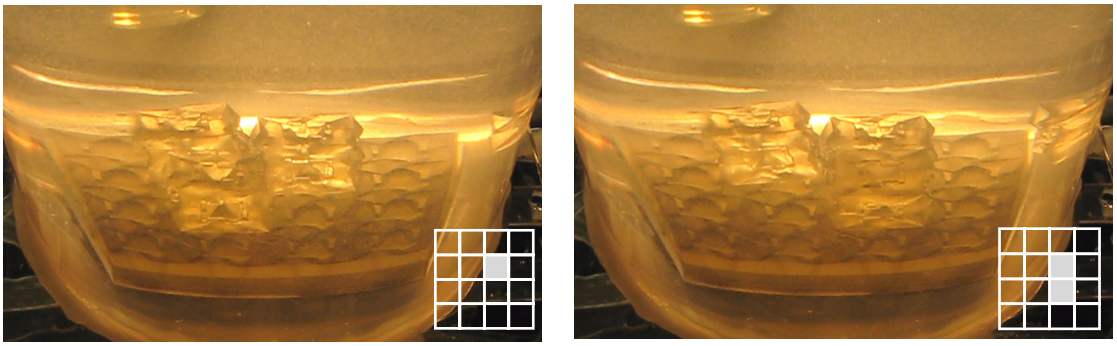
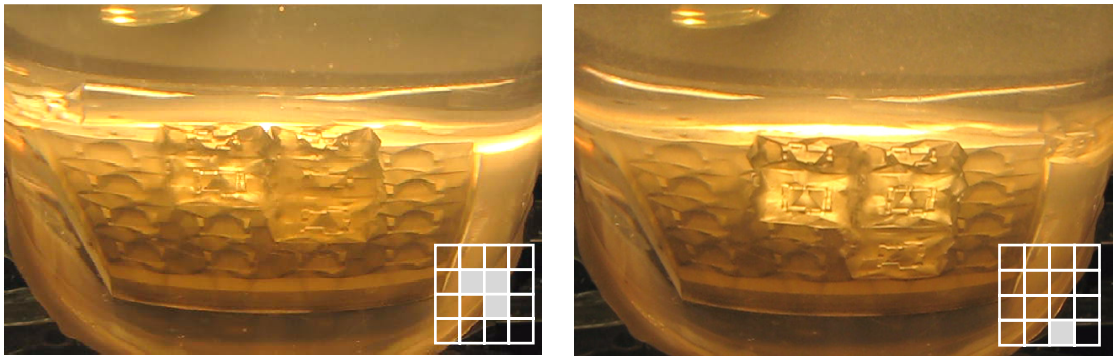


Figure 4.20. Two-module assembly. Images from an experiment in which two modules are attracted to the substrate and assembled. Rapidly switching the sink valves on and off causes the modules to vibrate and eventually snap into place. Closing the substrate valves allows ambient stochastic motion to carry the assembled pair away from the substrate. Filled squares in the grid overlay indicate occupied substrate locations.

It is also interesting to compare these experimental results with our model-based simulation of the stochastic fluidic assembly system presented in Chapter 5 [see also (Tolley, Kalontarov, Neubert, Erickson, & Lipson, 2010)]. In this work, we used a simulator to predict the time required to assemble one cube next to another on the centre of the bottom of a cylindrical fluidic chamber. The mean time to assembly was found to be 104 s in simulation, whereas here we found it to be 346 s experimentally. The discrepancy between these two values could be due to the fact that the cubes in simulation had flat sides, unlike those used in these experiments. This is consistent with the fact that in these experiments the majority of the time was spent vibrating the modules to induce them to latch together.



One Cube (t = 7m09s) Two Cubes (t = 7m29s)



Three Cubes (t = 7m56s) Standing Upright (t = 8m47s)

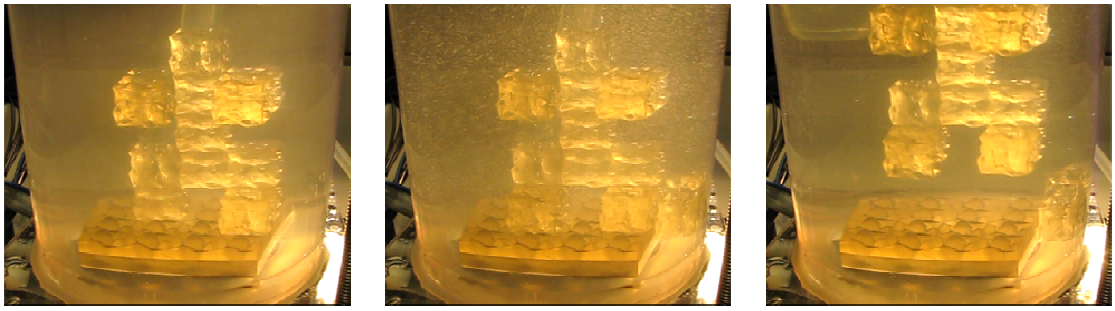
Figure 4.21. L-shape assembly. Images from an experiment in which three modules are attracted to the substrate and assembled. As with the two module case, valve switching causes the modules to snap into place. Once assembled, sinks are switched to sources to turn the L-shape upright. Filled squares in grid overlay indicate occupied substrate locations.

Table 4.3 Experimental Assembly Times

Experiment	Number of Experiments	Average Assembly Time (s)	Standard Error (s)
Two module assembly	5	346	60
Three module assembly	3	398	46
Structure repair	2	350	75
All experiments	11	341	37

Structure Repair Experiments

A third set of experiments was conducted to evaluate the ability of the system to repair damage on a complex, 14-module structure (Figure 4.22). A pre-assembled, 14-module anthropomorphic structure with a piece missing (representing damage) was inserted into the tank in a known position with the missing piece at the substrate. The substrate port at the location of the missing component was then opened as a sink in order to attract one of three surplus free modules. Once a module was attracted to the correct location, the valve switching frequency was adjusted as described previously to vibrate the module to the correct orientation and cause it to attach to the structure.



Insertion of Damaged Structure (t = 0s) | **Attraction of Module for Repair (t = 17s)** | **Removal of Repaired Structure (t = 3m28s)**

Figure 4.22. Structure repair. A damaged anthropomorphic assembly with right toe missing is repaired through stochastic fluidic assembly. Once the missing piece is replaced, the structure is removed from the assembly tank.

Results and Discussion

As seen in Table 4.3, it took an average of 350 s to replace the missing component on the structure inserted into the assembly chamber at a known position. Importantly, the vibration used to latch the replacement piece in place did not disassemble the original structure. However, in many cases the structure became dislodged from the substrate and had to be manually re-positioned. This is due to the fact that an insufficient number of controllable substrate ports were available to manipulate a

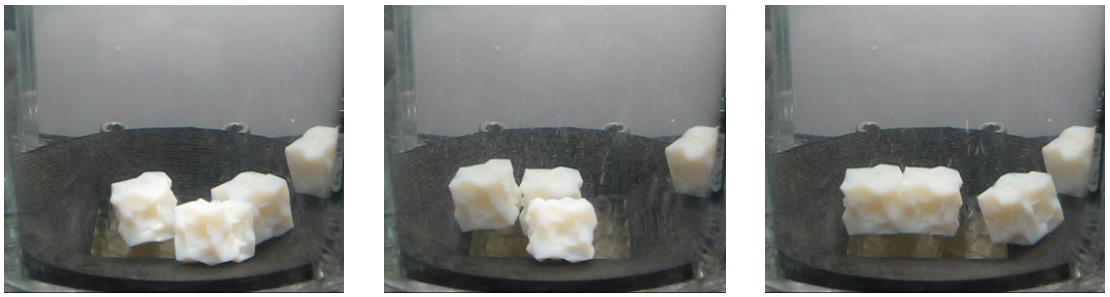
structure of the given size.

Robust 3-D Assembly

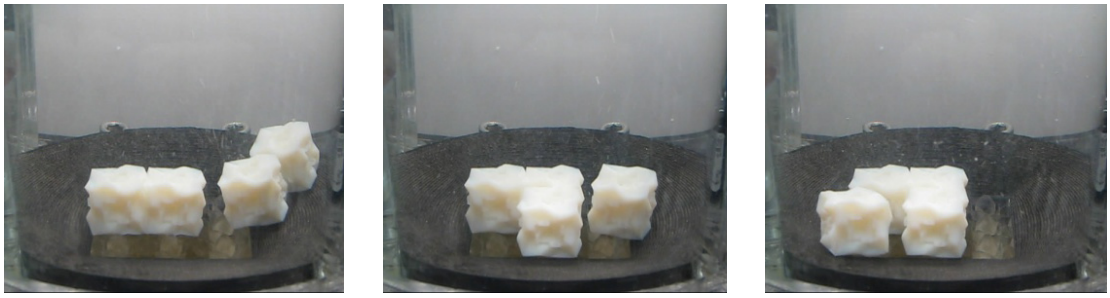
One of the key aspects of our stochastic fluidic assembly system is that the same forces used to manipulate modules can also be used to manipulate assemblies. Because of this feature, a 3-D structure can be assembled by releasing a 2-D structure assembled on the substrate and re-orienting it into an upright position. New modules can then be added at the substrate to form a non-planar 3-D structure. To demonstrate this concept, we performed a set of experiments in which we assembled an L-shape then reoriented it upright to add a fourth module (Figure 4.23). In order to evaluate the robustness of this approach, we repeated this process 10 times consecutively and recorded the assembly times for each stage of assembly.

These experiments were performed with Version 3 of our experimental system, with modules printed out of Objet's opaque VeroWhite material for improved visibility. Aside from improving the imaging of experiments (with the water lens and other improvements described above), the Version 3 system afforded an improved range of control that helped achieve robust assembly. In addition to four-state valving available for each substrate port, this system allowed continuous varying of source strengths from 0 (OFF) to 10 (ON) using pulse-width modulation. The challenge with this higher resolution of actuation is defining an assembly algorithm.

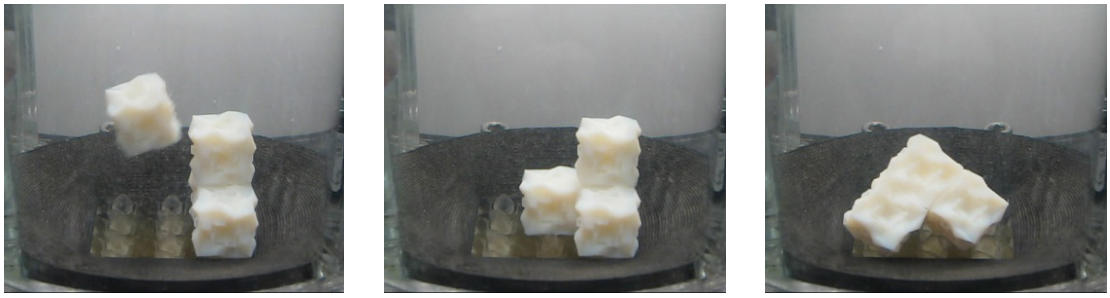
In order address this problem, we broke down the entire assembly process into general operations that take in a set of inputs and run until a certain condition is met. Thus we were able to tune open-loop valving sequences to achieve these operations, and limit the feedback needed for the identification of state transitions. This approach



Start | Stage 1: Align first cube | Stage 2: Align second cube

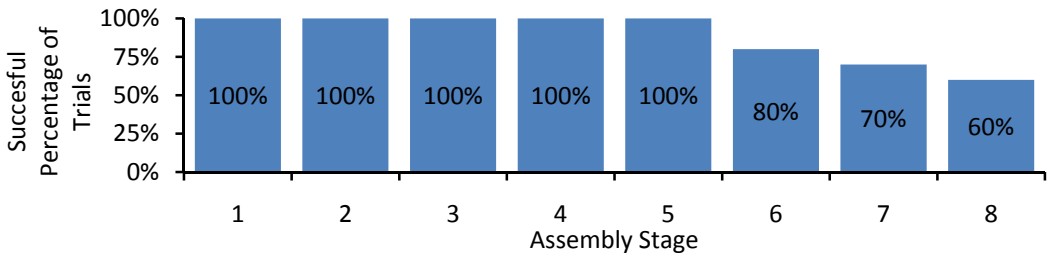


Stage 3: Latch pair | Stage 4: Align third cube | Stage 5: Latch elbow



Stage 6: Reorient | Stage 7: Align fourth cube | Stage 8: Latch and release

(a)



(b)

Figure 4.23. Robust 3-D Assembly. (a) Sequence of images from a 3-D assembly experiment displaying the eight required assembly operations. (b) Plot charting the percentage of 10 consecutive trials that achieved each of the assembly stages shown in (a). The modules seen here are printed out of Objet's VeroWhite material for improved visibility.

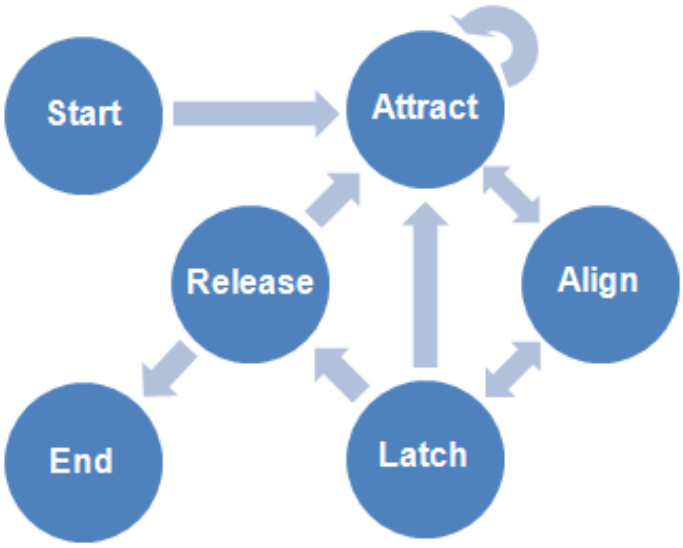


Figure 4.24. Finite state machine (FSM) for assembly control. Assembly consists of a sequence of four fundamental operations corresponding to the four main states seen here. In the experiments, control for each state was mostly automated while user feedback was used to identify operation completion and initiate state transitions.

has two advantages: First, it makes the problem of manual control of assembly tractable since turning 18 knobs in real time to achieve repeatable assembly is difficult at best (see Figure 4.15). Second, the more that human feedback can be limited while developing such strategies, the easier it is to automate the process later.

In essence, the approach we are describing is a finite-state-machine approach (Figure 4.24). We found that we were able to achieve assembly using only four operational states: *Attract*, *Align*, *Latch*, and *Release*. Each one of these states takes input on the current and/or desired state of the system and runs a sequence of valve switches to achieve a goal state. In these experiments, human feedback was used to determine when either a goal state was achieved or the operation failed, and to switch to the next state accordingly. The arrows on Figure 4.24 show the potential transitions possible between each pair of states. For example, after the first module is attracted, the state returns to *Attract* to obtain a second module. The state then transitions to

Align to obtain a new one. If alignment is successful, the state transitions to latching. However, if one of the modules is accidentally released from the substrate during alignment, the state returns to *Attract*. Table 4.4 gives a brief description of the open loop valving sequences used for each one of these stages.

Table 4.4. Finite-state-machine open-loop assembly control algorithms

State	Inputs	Algorithm
Attract	Desired cube locations	• Open sinks at desired cube locations • Open sources of strength A_{min} at all other ports • Loop through sources one at a time, increasing their strengths to Amax, then back down to A_{min} • Done when cubes have been attracted to each desired location
Align	Locations of mated and unmated cubes	• Open sinks at cube locations • Also open source at mated cube location and increase its strength slowly until cubes align • Done when cubes are aligned (go to latch), or a target location is no longer occupied by a cube (go to attract)
Latch	Cube locations	• Open sinks at cube locations • Open sources at cube locations • Increase source strengths slowly to vibrate cubes until cubes latch (go to attract or release as dictated by assembly plan), or become misaligned (go to align)
Release	Cube locations	• Open sources at cube locations • Slowly increase sources to release structure from substrate • For reorientation, attract to new location (attract), otherwise assembly is done

Results and Discussion

Figure 4.23(a) presents a sequence of images from a robust 3-D assembly experiment that illustrate the 8 stages required for the assembly of the target structure shown (note that these do not all correspond to the states discussed above since some stages comprise multiple states). The depicted experiment is one from a series of 10 performed consecutively. Figure 4.23(b) shows the percentage of these experiments that made it successfully to each assembly stage. For example, while 6 of 10 experiments completed the target structure, 8 of 10 made it to the reorientation stage but failed to complete the alignment of the fourth cube successfully. A failure was defined as an error that could not be fixed without returning to the first stage. Otherwise, repeated attempts were allowed after unsuccessful operations (for example, if aligned cubes became unaligned during latching).

Obviously, this definition of failure is somewhat arbitrary. In theory, any sequence of operations could be repeated until the desired result was achieved. For this reason, a more appropriate measure of assembly efficiency is the time required to complete each stage. Figure 4.25 summarizes these assembly times for the robust 3-D assembly experiments. Each dataset represents one experiment, with the vertical axis representing the ordered assembly stages and the horizontal axis denoting the corresponding experimental time. The mean time to the final assembly stage was 442 s, however the median was 339 s. This is because the mean is largely affected by one outlier experiment that took much longer than the others.

Looking at the individual experiment times, we see that most experiments spent a similar amount of time on each stage. However, two experiments took a relatively long time to get a second module aligned and one also took a long time to achieve successful reorientation. Comparing the average assembly times (the dotted line) with the previous experiments (Table 4.3), we see that these experiments, with the

upgraded Version 3 experimental system, achieved assembly much more quickly. In these experiments, pair assembly took an average of 165 s, versus 346 s previously. Similarly, the time for assembly of three modules was reduced from 398 s to 281 s.

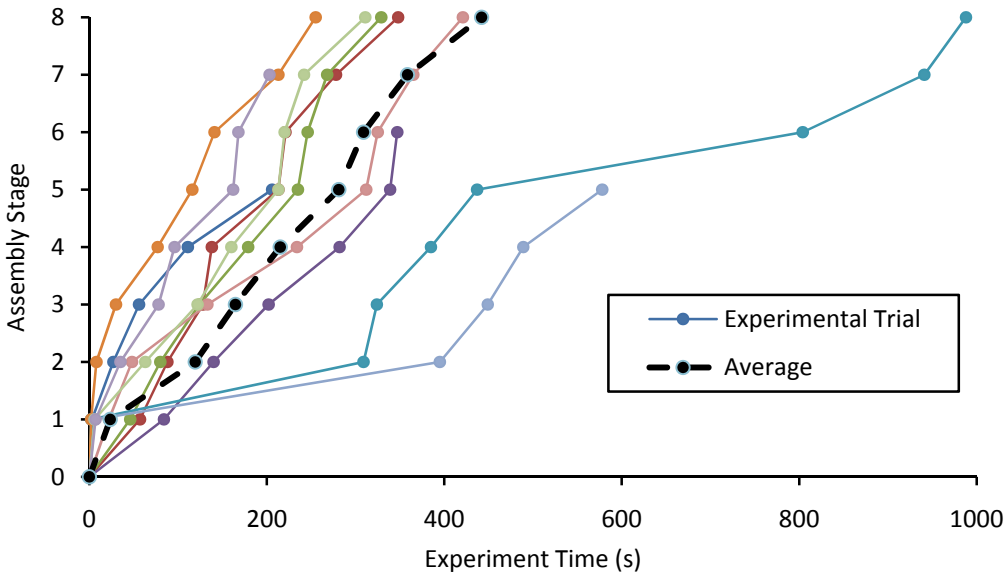


Figure 4.25. Robust assembly experiment times. Plot of assembly times of 10 consecutive 3-D assembly experiments. Each data series represents one experiment. Data points indicate the time elapsed since the start of the experiment when the corresponding assembly stages were achieved. The *average* dataset corresponds to the average time taken to achieve each assembly stage.

Hierarchical 3-D Assembly

In Chapter 3 we discussed the use of hierarchical assembly as a means of reducing assembly times and the cost of errors for stochastic fluidic assembly. This advantage is not unique to 2-D or micro-scale systems. For similar reasons, the 3-D stochastic fluidic assembly system under consideration here also benefits from hierarchical assembly. In fact, one could argue that hierarchical assembly, and the parallel operations it allows, is the only practical way to achieve stochastic fluidic assembly of a structure composed of many modules, since serial approaches take a prohibitively

long time.

In order to demonstrate our system's ability to achieve hierarchical assembly, we conducted proof-of-concept experiments to assemble structures that form the building blocks for subsequent assembly. A key aspect of the stochastic fluidic assembly approach that enables hierarchical assembly is the fact that our system is able to manipulate assemblies in the same way it manipulates the individual modules. Thus, in these experiments, the goal was to demonstrate the assembly of sub-structures comprising two modules, then to form both planar and non-planar four-module structures from these sub-assemblies. These experiments were conducted using a similar approach to that described in the previous section, with a FSM control algorithm supervised by a human operator using visual feedback. The advantages of hierarchical assembly rely largely on the ability of the system to assemble structures in parallel. Thus, the second set of experiments aimed to assemble two 3-D structures simultaneously (at least as much as possible giving the limitations of human feedback).

Results and Discussion

Figure 4.26 shows frames from a video of an experiment demonstrating the hierarchical assembly of a planar L-shape composed of four modules. A pair of modules was first assembled at the substrate, then rejected into the flow. This process was then repeated for a second pair. The two pairs were then attracted to the substrate in the target configuration and induced to latch using a similar approach to the previous serial assembly experiments. Finally, the four-cube structure was released from the substrate and was found to remain assembled. A greater number of modules were used in this experiment than required for the final structure in order to accelerate assembly.

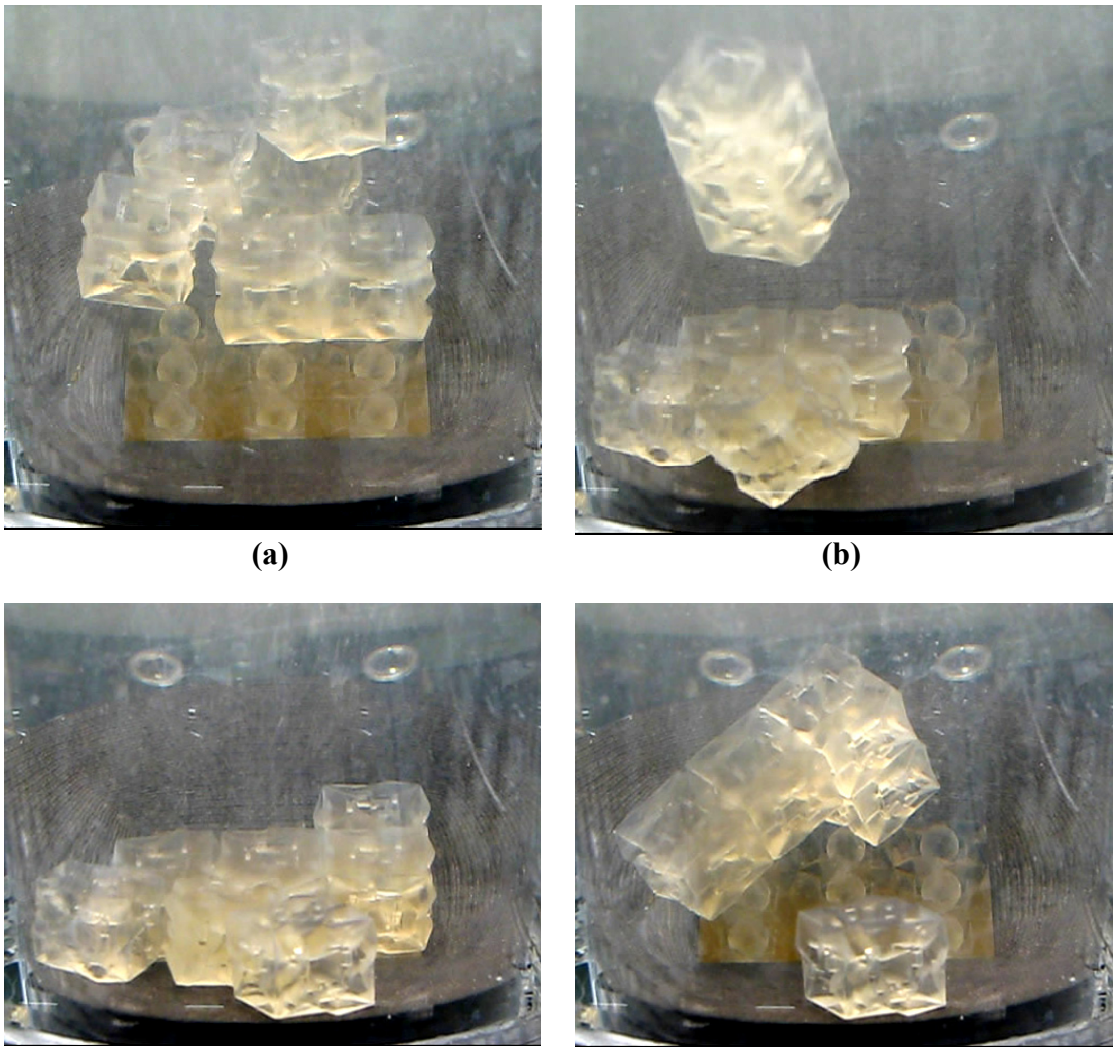


Figure 4.26. Hierarchical assembly of an L-Shape structure. (a) First two modules are assembled on the substrate. (b) The assembled pair is then released back into the fluid volume while a second pair is assembled on the substrate. (c) In the second stage of hierarchical assembly, these two pairs are then attracted to the substrate and latched together to form the target structure. (d) The final structure is then released from the assembly substrate. Excess modules do not form part of the final structure

A second set of experiments, illustrated by Figure 4.27 and Figure 4.28, demonstrate the parallel hierarchical stochastic fluidic assembly of two 3-D structures. As with the experiment presented in Figure 4.26, the approach was to first assemble individual modules into pairs, then to assemble these pairs into four-module structures.

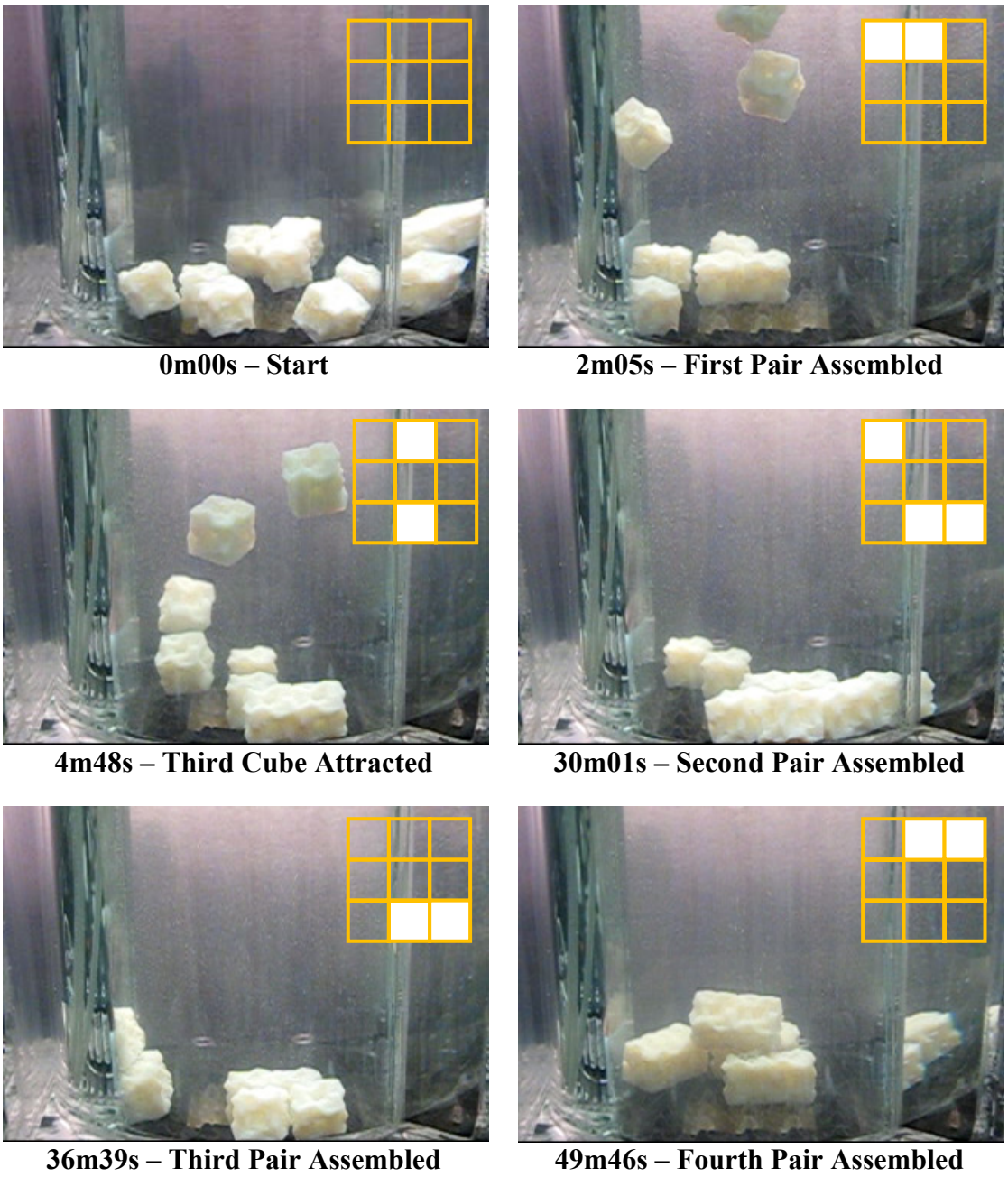
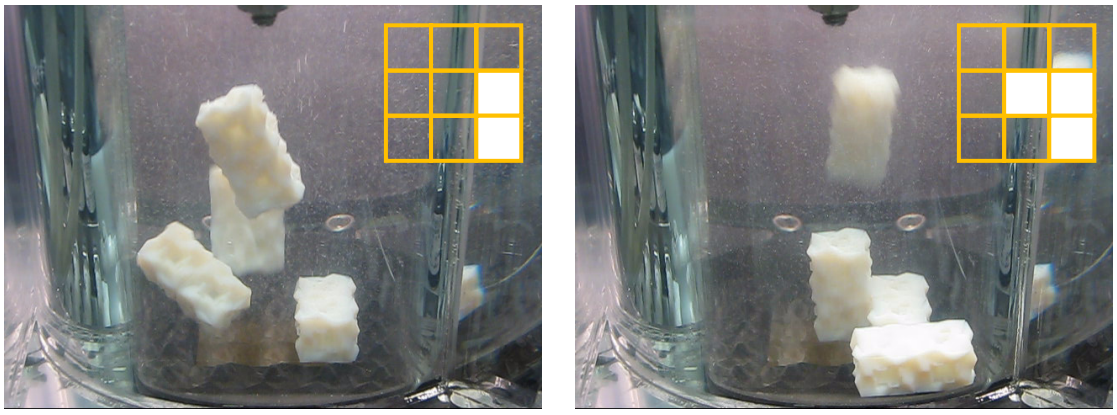
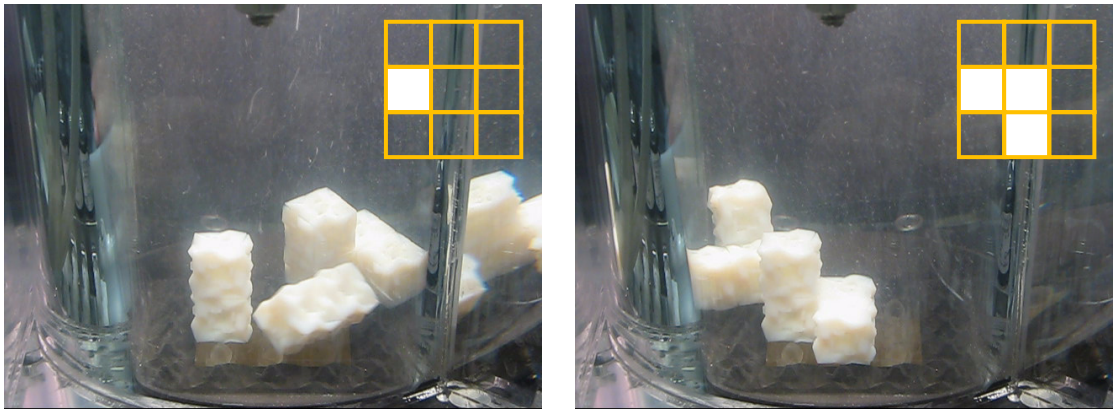


Figure 4.27. First stage of parallel hierarchical 3-D assembly. Selected frames from an assembly experiment representing the first stage of the parallel hierarchical assembly of two 3-D structures. During this stage, the stochastic fluidic assembly approach is used to assemble eight individual modules into four pairs of modules, to be assembled into 3-D structures in the second stage (see Figure 4.28). The grid on the upper-right hand corner of each image indicates the locations on the 3 × 3 array of substrate ports that are occupied by modules. Pair assembly occurs in simultaneously on the top and bottom rows of the array.



0m12s – First Pair Attracted **2m05s – First Quad Assembled**

(a)



10m52s – Third Pair Attracted **11m37s – Second Quad Assembled**

(b)

Figure 4.28. Second Stage of Parallel Hierarchical 3-D Assembly. Two separate experiments demonstrate the second stage of the parallel hierarchical assembly of two 3-D structures. (a) The first experiment begins with the four cube pairs assembled in the first stage (see Figure 4.27), and assembles two of the pairs into a 3-D structure. (b) The second experiment subsequently assembles the remaining two pairs into a second, identical 3-D structure. The grid on the upper-right hand corner of each image indicates the locations on the 3 × 3 array of active substrate ports that are occupied by modules.

However, in this case, the four-module structures were non-planar, and two of them were assembled in parallel. Non-planar structures were achieved in a method similar to that presented in the experiments described in the *Robust 3-D Assembly* section above. Specifically, planar sub-assemblies (the pairs) were assembled on the substrate, then later attracted to the substrate in an upright position, for further assembly.

However, in this case, sub-structures were added to the upright structure as opposed to individual modules, to demonstrate hierarchical assembly. Figure 4.28(a) shows the addition of an upright pair to a horizontal one, while in Figure 4.28(b), a horizontal pair is added to an upright one.

Parallel hierarchical assembly was achieved by starting with enough modules in the tank for the assembly of two 3-D structures (i.e. eight modules). Unlike the experiment shown in Figure 4.26, excess modules were not used in this case in order allow for experimental imaging (however, this likely contributed to the long times required for assembly). Pairs of modules were then assembled simultaneously on the top and bottom rows of the substrate (Figure 4.27). However, because of the current human-supervised approach, any steps that required the operator's attention (i.e. initiating state transitions as described above on the *Robust 3-D Assembly* section) had to be processed serially. Additionally, the 3 × 3 size of the substrate limited the ability of the system to conduct the second stage of assembly in parallel (see Figure 4.28). Nonetheless, these experiments demonstrate the ability of the system to assemble multiple copies of 3-D assemblies and subassemblies in parallel in order to realize the advantages of hierarchical assembly.

Conclusions

In this chapter, we have presented an experimental system for the 3-D assembly of modular robot components. This system aims to increase the scalability of modular robots to large numbers of small modules. Our approach to this end is to reduce the complexity of the modules required for assembly as much as possible in order to maximize the amount of resources a module is able to spend on achieving its functions within the assembled robot. We do this by assembling the modules in a stochastic fluidic environment which takes care of module transportation. We then manipulate

the fluid flow at an active assembly substrate using external valving to direct component assembly. We have presented three generations of our experimental system to achieve this goal, and explained the design choices made for the system modules, hardware and software. In addition, we have tested a number of the functionalities of the system and evaluated how they compare experimentally to their intended functions.

We then described five sets of experiments that demonstrate various aspects of programmable 3-D stochastic fluidic assembly. The first set of experiments demonstrated the capability of the system to attract a module to a specific site on the assembly substrate and manipulate it reliably, despite the stochastic nature of the fluidics. The second set of experiments demonstrated the capability of the system to assemble two- and three- module structures by inducing the modules to latch together at the assembly substrate. In the third set of experiments, a damaged humanoid structure with a piece missing was inserted into the assembly tank, and the missing piece was replaced. The repaired structure could then be removed from the assembly tank. For the fourth set of experiments, a target non-planar structure was fabricated by assembling a structure on the planar substrate, then reorienting the assembly out of the plane to add another module. The control approach, based on a FSM algorithm, was described, and found to assemble the desired structure 6 out of 10 consecutive experiments in an average of 442 s. Finally, the fifth set of experiments demonstrated the system's ability to assemble structures hierarchically, a function that has the potential to greatly reduce assembly times and the cost of errors.

The work described in this chapter represents the first steps toward a large-scale system where subassemblies are manufactured on many separate assembly substrates and brought together through a combination of directed and stochastic processes to form complex machines. This assembly approach is reminiscent of what occurs in

nature and would likely share some of natural assembly's advantages, such as versatility, robustness, and evolvability, as well as its challenges, such as inefficiency and unpredictability. We hope that future iterations of this system, implementing closed-loop feedback as described in this chapter, will take further steps towards this goal.

PART III: SIMULATION

CHAPTER 5: STOCHASTIC FLUIDIC ASSEMBLY SIMULATION

Abstract

Modular robotic systems typically assemble using deterministic processes where modules are directly placed into their target position. By contrast, stochastic modular robots take advantage of ambient environmental energy for the transportation and delivery of robot components to target locations, thus offering potential scalability. The inability to precisely predict component availability and assembly rates is a key challenge for planning in such environments. Here, we describe a computationally efficient simulator to model a modular robotic system that assembles in a stochastic fluid environment. We first calibrate the simulator using both high-fidelity computational fluid-dynamics simulations and physical experiments. We then use this simulator to study the effects of various system parameters on the speed and accuracy of assembly of topologically different target structures.

Introduction

Modular self-reconfigurable robotic systems have the potential to adapt to new environments and tasks by changing the connectivity of their constituent modules to transform their morphology (Yim, et al., 2007; Fitch & Rus, 2003). This capability could result in a versatile system that can accomplish unforeseen goals, repair itself when damaged, efficiently reuse components, and self-replicate. These remarkable advantages, however, come with severe challenges in the mechanical design and control of the system and its modules. Previous work on modular self-reconfigurable robot systems has addressed many of these challenges: A variety of system designs

have been proposed, including mobile modules with chain (Fukuda, Nakagawa, Kawauchi, & Buss, 1988; Yim, Zhang, & Duff, Modular robots, 2002; Castano, Behar, & Will, 2002; Zykov, Mytilinaios, Desnoyer, & Lipson, 2007) and planar or 3-D lattice (Murata, Yoshida, Kamimura, Kurokawa, Tomita, & Kokaji, 2002; Pamecha, Chiang, Stein, & Chirikjian, 1996; Goldstein, Campbell, & Mowry, 2005) connectivity. Such approaches require complex mechanisms and high energy budgets that are difficult to scale to small dimensions and large numbers. An alternative approach sidesteps the demands of module locomotion by allowing the modules to move freely in a stochastic environment and by controlling the connectivity only when modules come into contact (Griffith, Goldwater, & Jacobson, 2005; Klavins E. , 2007; Gilpin, Kotay, Rus, & Vasilescu, 2008; White, Kopanski, & Lipson, 2004; White, Zykov, Bongard, & Lipson, 2005). This biologically inspired stochastic assembly approach forms the basis of the work presented here.

Previous work has examined various aspects of the design and control of robotic stochastic assembly systems. Inspired by the self-assembly research (Whitesides & Grzybowski, 2002; Penrose & Penrose, 1957) these systems generally add the ability to control their configurations on the fly. White *et al.* (2004) first demonstrated the stochastic self-assembly and reconfiguration of triangular modules on an air table and suggested that simple assembly strategies could lead to dramatically different scalability. These principles were then repeated in 3-D (White, Zykov, Bongard, & Lipson, 2005; Zykov & Lipson, Experiment design for stochastic three-dimensional reconfiguration of modular robots, 2007). Griffith *et al*. (2005) demonstrated the self-replication of a 2-D template string from electromechanical parts moving about stochastically on an air table. When the parts come into contact, they latch together, communicate with one another, and decide whether to disengage the latches. Klavins (2007) designed a similar 2-D system of "programmable parts" along with a method of

modeling the connectivity of these parts using graph grammars. A control scheme was also proposed in which the various interactions are viewed as chemical reactions with parameters that can be tweaked to achieve desired target structures. Gilpin *et al*. (2008) have taken alternative self-disassembling approach that begins with a lattice of modules that communicate and establish connectivity as required to form a desired structure while unnecessary components are released stochastically.

In Part I of this thesis, we have presented our previous work on the assembly of 2-D structures from 500-μm-scale silicon components in a fluidic environment (Tolley, Krishnan, Erickson, & Lipson, 2008). These components had a passive latching mechanism and assembled deterministically into arbitrarily defined structures with open-and closed-loop fluid control. We also previously demonstrated experimental 3-D assembly from robotic cube-shaped modules in a fluidic environment (White, Zykov, Bongard, & Lipson, 2005). However, the relatively large scale of this system (10 cm) led to slow assembly rates and made it difficult to demonstrate the experimental assembly of more than few components. This experimental work was complemented by a basic 3-D simulator that was used to examine some aspects of 3-D stochastic assembly. This simulator, however, included no fluidic forces.

One of the major challenges in self-reconfigurable modular robotics has been scaling up the number of modules. The capability of a modular robotic system to realize its advantages over traditional systems is based largely on its ability to assemble large numbers of components with a fine resolution; however, systems composed of more than ~50 modules have yet to be demonstrated (Yim, et al., 2007). In order to increase the resolution of current systems, it will be necessary to further reduce their modules' sizes. Despite the reduction in module complexity due to stochastic assembly approaches, current limitations of microfabrication technologies

(e.g., their 2-D nature) make the manufacture of 3-D robotic modules a difficult problem. Nonetheless, we believe that it should be possible to reduce all of the necessary components of a modular robotic system based on our fluidic assembly approach to fit inside a 1-cm cube. For this reason, we aim to develop a system of stochastic, fluidically assembled modules of this scale.

In this chapter, we present a custom 3-D simulator to support this experimental effort. While a simulator is no substitute for a physical system, it does enable us to explore the large space of possible system parameters and assembly strategies much more efficiently. The challenge with solving mixed fluid–rigid-body systems is the high cost of computation. Our goal here is to make sufficient simplifications to make the problem tractable while still obtaining meaningful results. In order to gain confidence in our simulator, we compare its results with those of a commercial computational fluid dynamics (CFD) package and with experimental results for specific test cases. We then use the simulation to examine the effects of different system parameters on assembly dynamics. Based on these results, we recommend system parameters for a fluidic assembly system.

Fluidic Self-Assembly Concept

At small scales, biological structures assemble themselves primarily in fluidic environment taking advantage of random Brownian motion as a component transportation mechanism. Inspired by this example, our approach to self-reconfigurable modular robotics involves unpowered cubes that rely on ambient stochastic fluid motion for transportation (White, Zykov, Bongard, & Lipson, 2005; Zykov & Lipson, 2007). Fluidic forces are additionally used to accelerate assembly by attracting cubes to where they are needed. Finally, a bonding force is used to hold the cubes together in the final structure.

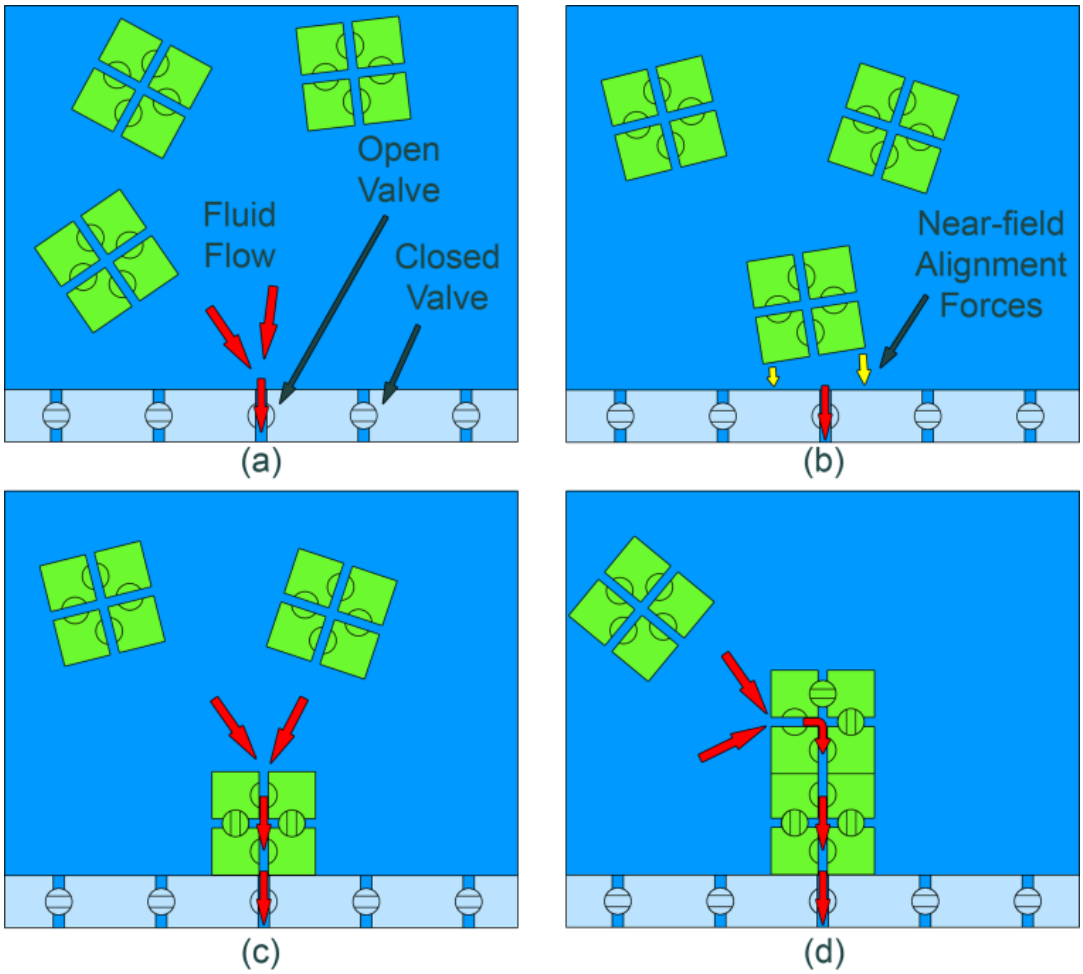


Figure 5.1. Fluidic Self-Assembly Concept. (a) Fluid flow (which is indicated by arrows) into a substrate attracts a nearby module. (b) Once a module passes within close proximity of the target location, near-field forces (e.g., magnets) cause the module to align and attach. (c) Once attached, the module draws power from the substrate to activate on-board valves and redirect fluid flow through internal channels, thereby (d) attracting new modules at desired locations. This process continues layer-by-layer until the structure is complete.

Structure formation begins by opening a sink on a growth substrate in order to attract nearby cubes (see Figure 5.1). When a cube falls within the basin of attraction of the sink, it is pulled toward the sink where geometric interactions cause it to align with the growth substrate and a bonding mechanism activates to hold the cube in place. Once attached, the cube is able to draw power from the substrate to activate

internal valves, thereby closing off internal channels as required to connect the bonded face with any number of exposed faces. This effectively moves the sink from its original location to one or more surfaces of the attached cube to attract new cubes to these locations. The target system is thus "grown" by repeatedly opening sinks and by attracting and bonding cubes. Reconfiguration is achieved by deactivating the bonds to unwanted modules and allowing ambient fluid motion to carry them away while attracting components to new locations as required.

Simulation

The goal of this study was to develop a computationally efficient simulator to aid in the design and operation of a system based on the stochastic fluidic assembly concept. Specifically, our goal was to develop a simulator that is capable of predicting the assembly rates and completion percentages for an arbitrary structure using various assembly approaches and system parameters. During the experimental system design phase, a simulator allows the exploration of various system parameters to inform the design and avoid unnecessary iterations. Additionally, even after completion of the experimental system, an accurate simulator allows experimentation with strategies and scenarios that would be impossible or too costly to test physically.

Choice of Simulation Method

In order to achieve our goals, we required a simulator that was as accurate as possible while maintaining computational efficiency. Using a full CFD simulation coupled with a rigid body solver would have been the most accurate approach to predicting the motion of the components in the assembly tank. However, this approach would also have been prohibitively expensive. For example, simulations of the motion of a single cube approaching a sink from a distance of two cube lengths using the CFD

software FLOW-3D took approximately 4.8 h to solve. Our initial goal was to simulate the assembly of a structure composed of approximately 100 cubes and to repeat the simulation for many different parameters and assembly strategies. Even with many computers working in parallel, it was apparent that CFD simulations would not be a tractable option. We thus decided to develop a simulation that would capture the assembly dynamics of the system without getting lost in the details of solving the fluid flow.

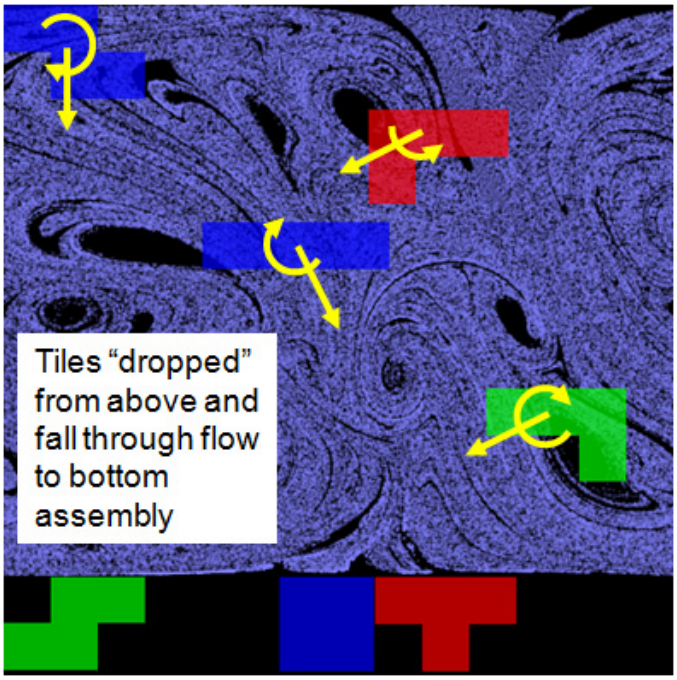
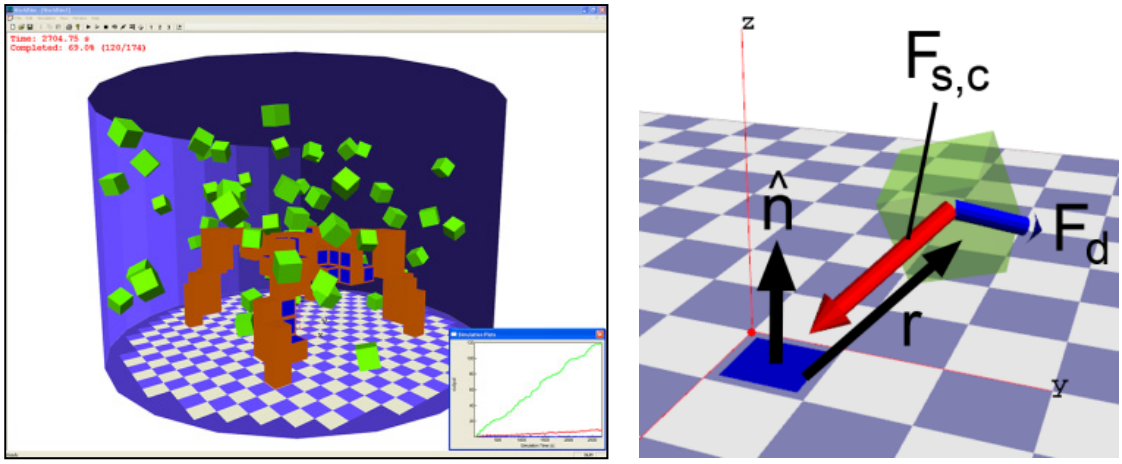


Figure 5.2. Stable fluids-based simulator. Initial attempts at developing a computationally efficient fluids simulator based on the *Stable Fluids* approach (used here to implement a game of *Fluidic Tetris*) convinced us to pursue an alternate approach.

Our initial approach to a computationally efficient fluid simulation was based on an approach developed by Joe Stam for the efficient animation of fluid flows in computer graphics (Stam, 1999). However, preliminary investigations determined that it would take a great deal of customization to adapt this approach to the simulation of

our programmable matter system. It would have been necessary to extend the available code from two to three dimensions and to implement an efficient rigid body solver to account for the fluid-component and component-component interactions. A 2-D version of a *Stable Fluids*-based simulator used to implement a "fluidic tetris" can be seen in Figure 5.2.



(a) (b)

Figure 5.3. Stochastic fluidic assembly simulator. (a) To achieve computationally efficient simulation of our modular robotic system, we wrote a custom simulator in C++ using the ODE libraries. (b) Simplified fluidic forces were added to the ODE rigid-body simulation to model the forces applied to the modules. A module is shown here transparent with arrows representing these forces. The arrow labeled $F_{s,c}$ represents the force exerted on the module by fluid exiting the assembly tank through the open sink (which is represented by a dark square). F_d is the fluid drag force resisting the motion of the cube. $\hat{n}$ is a normal to the sink, and r is a vector from the sink to the cube.

Instead of starting with an approximate fluids solver and integrating a custom rigid body solver, we decided to follow the opposite approach. We wrote our fluidic assembly system simulator [see Figure 5.3(a)] in C++, using the Open Dynamics Engine (ODE) (Smith., 2004)—a stable, open-source, adaptable, computationally efficient rigid-body solver—for simulation of cube motion and collision detection. We then added simplified fluidic forces to model the effects of agitation and fluid drag, as

well as nearfield alignment forces, and the capability to lock cubes together and to the substrate. By adjusting the physical properties of the system (such as friction coefficients, viscosity, etc.) in ODE and the custom forces, we were able to simulate a wide variety of system configurations. We then added a framework to load target shapes and to open and close fluid sinks following various assembly strategies (see Chapter 6 for details on assembly strategy development).

Fluid Forces Model

Simplified fluidic forces were applied to the cubic components of our simulated modular robotic system in order to approximate the forces that the cubes would experience in experiment. We calculated these forces based on the velocities of the cubes and their positions relative to any open sinks. We also added a random component to model fluid agitation. The first two forces—the force of a sink on a cube and the fluidic drag force resisting cube motion—are represented by the forces labeled $F_{s,c}$ and F_d, respectively, in Figure 5.3(b). This is a frame from a simulation video with a module being attracted to a sink.

We can derive the equations for these forces starting with the force caused by fluid moving with respect to a cube as follows:

$$F_c = \frac{\rho}{2} C_D A v^2 \tag{5.1}$$

where F_c is the force of the fluid flow on the cube, ρ is the density of the fluid, C_D is the drag coefficient for a cube in a flow, A is the area of a face of the cube, and v is the relative velocity of the cube with respect to the fluid. In the case of Stokes' flow, we have

$$C_D = \frac{24}{\mathrm{Re}} = \frac{24\,\mu}{\rho v d} \tag{5.2}$$

where Re is the Reynolds number of the flow, d is the characteristic length (i.e., side

length) of the cube, and μ is the viscosity of the fluid. Substituting (2) into (1), we have the following equation:

$$F_c = 12\mu d v \tag{5.3}$$

From continuity, we know that the volumetric flow rate through a hemisphere with radius r centered on a single sink draining fluid from a tank is equal to the flow rate through the opening of the sink itself

$$U_0 A_0 = U_r A_r \tag{5.4}$$

where U_r and U_0 are the velocities of the fluid at the hemisphere and sink, respectively, and A_r and A_0 are the areas of the hemisphere and sink opening, respectively. We can thus relate the velocity of the flow at a radius r away from a sink to the velocity through the sink opening with radius R_0

$$U_r = \frac{U_0 \pi R_0^2}{4\pi r^2} = \frac{U_0 R_0^2}{4r^2} \tag{5.5}$$

Now, the relative velocity of the cube with respect to the surrounding flow at a radius r away from a sink is given by

$$v = U_r - v_c \tag{5.6}$$

where v_c is the velocity of the cube with respect to the inertial frame.

Substituting (5) into (6), and the result into (3) yields

$$F_c = 12\mu d\left(\frac{U_0 R_0^2}{4r^2} - v_c\right) = 3\mu d \frac{U_0 R_0^2}{r^2} - 12\mu d v_c \tag{5.7}$$

The first term in (7) is the effect of the sink on the cube, whereas the second term is the effect due to the motion of the cube. In the case where k sinks are open and connected to the same outlet with flow velocity U_0, this force gets divided by k. Assuming further that a sink only affects cubes in front of its face, we get the

following equation for the sink force on a cube ($F_{s,c}$):

$$F_{s,c} = \begin{Bmatrix} 3\mu d \dfrac{U_0 {R_0}^2}{kr^2}, \hat{n} \cdot r > 0 \\ 0, \hat{n} \cdot r \leq 0 \end{Bmatrix} \tag{5.8}$$

where $\hat{n}$ is a unit normal vector pointing away from the sink. Thus, we have the following equation for the drag force due to the cube motion with respect to the inertial frame (F_d):

$$F_d = -12\,\mu d v_c \tag{5.9}$$

Assuming that sink effects superimpose linearly, we sum the contributions $F_{S_i,C}$ of each individual sink S_i to obtain the overall fluidic force on the cube due to k sinks:

$$F_C = \sum_{i=1}^{k} F_{S_i,C} + F_d \tag{5.10}$$

$$F_{S_i,C} = \begin{Bmatrix} 3\mu d \dfrac{U_0 {R_0}^2}{k {r_i}^2}, \hat{n}_i \cdot r_i > 0 \\ 0, \hat{n}_i \cdot r_i \leq 0 \end{Bmatrix} \tag{5.11}$$

where r_i is the position of the cube with respect to the ith sink, and $\hat{n}_i$ is a unit normal vector perpendicular to the face of the *i*th sink.

CFD and Experimental Validation of Module Attraction

We used a CFD software package, i.e., FLOW-3D (Flow Science, Santa Fe, NM, 2000), and an experimental setup to validate the fluid forces model described in the previous section in the case of module attraction (see Figure 5.4). We examined the test case of a single module being attracted from a distance of two module lengths to a sink on top of an assembled two-module structure from various approach angles, which are defined by their angle θ from the sink's normal. This test case was constructed in both simulators and in the experimental setup with the specifications listed in Table 5.1.

Table 5.1 Test-Case Specifications

Quantity	Value	Quantity	Value
Module Length	1 cm	Assembly Fluid	Water
Module Buoyancy	Neutral	Temperature	20°C
Sink Radius	0.1 cm	Density	0.001 kg/cm^3
Tank Radius	6 cm	Dynamic Viscosity	0.001 Pa·s
Tank Height	16 cm	Average Relative Velocity Between Cube and Fluid	0.005 cm/s
Initial Module Distance	2 cm	Reynold's Number	0.5
Sink Flow Velocity	10 cm/s	Sink Flow Rate	0.31 mL/s

FLOW-3D provides a good basis for comparison as it allows the modeling of dynamic fluid flows and their interactions with mobile rigid bodies. By coupling the fluid and rigid-body motions, the software simulates the motion of the rigid bodies due to hydrodynamic forces by numerically solving the Navier–Stokes equations. However, this process is very computationally expensive. We reduced the amount of unnecessary computation by solving only over a limited volume around the rigid bodies where velocities were likely to change, thus setting the surrounding top and side boundaries to the continuity condition. The total mesh volume was thus 2 cm × 4 cm × 4 cm. Nonetheless, the CFD simulations took approximately 4.8 h to complete. By comparison, the longest custom simulations took less than 10 s on a comparable personal computer.

For our experimental setup, we avoided the difficulties of achieving neutral buoyancy and precise initial positioning in three dimensions by attracting a cube in the plane on the bottom of a tank of water [see Figure 5.4(d)]. A position guide under the transparent bottom was used to place the cubes in their initial locations, at which point

water was pumped out through the sink. The resulting cube motion was captured from above with a high-speed camera. For each trial, the entire cube motion was divided into four equal time intervals, and the position of the cube was extracted from the beginning and the end of each interval. The denser-than-water cubes were weighted as closely as possible to neutral buoyancy in order to reduce friction with the tank bottom. However, it was still necessary to increase the sink flow rate much higher than in simulation (to 763 cm/s) to initiate cube motion.

It should also be noted that for this set of simulations and experiments, since our goal was to validate our sink-attraction model, we did not include any sort of *near-field force* to align cube faces as they approach one another. While such a force was found to have a significant effect on assembly rates (see *Near-Field Force* below), we felt that adding such a mechanism to the present comparison would overly convolute the results.

The results of the attraction-model -validation comparison can be seen in Figure 5.4. Figure 5.4(a)–(c) shows superimposed images of the cube's positions at regular intervals for the $\theta = 30°$ case from the custom simulation, CFD simulation, and experiments, respectively. For the simulations, each image represents a 10 s interval, while in the experiment, the interval between cube images is 0.2 s (since the higher flow rate in experiments resulted in much quicker cube motions). The cube in the custom simulation can also be seen to move more slowly than that in the CFD simulation. This suggests that our fluid forces model underestimates the strength of the hydrodynamic forces applied to the cube.

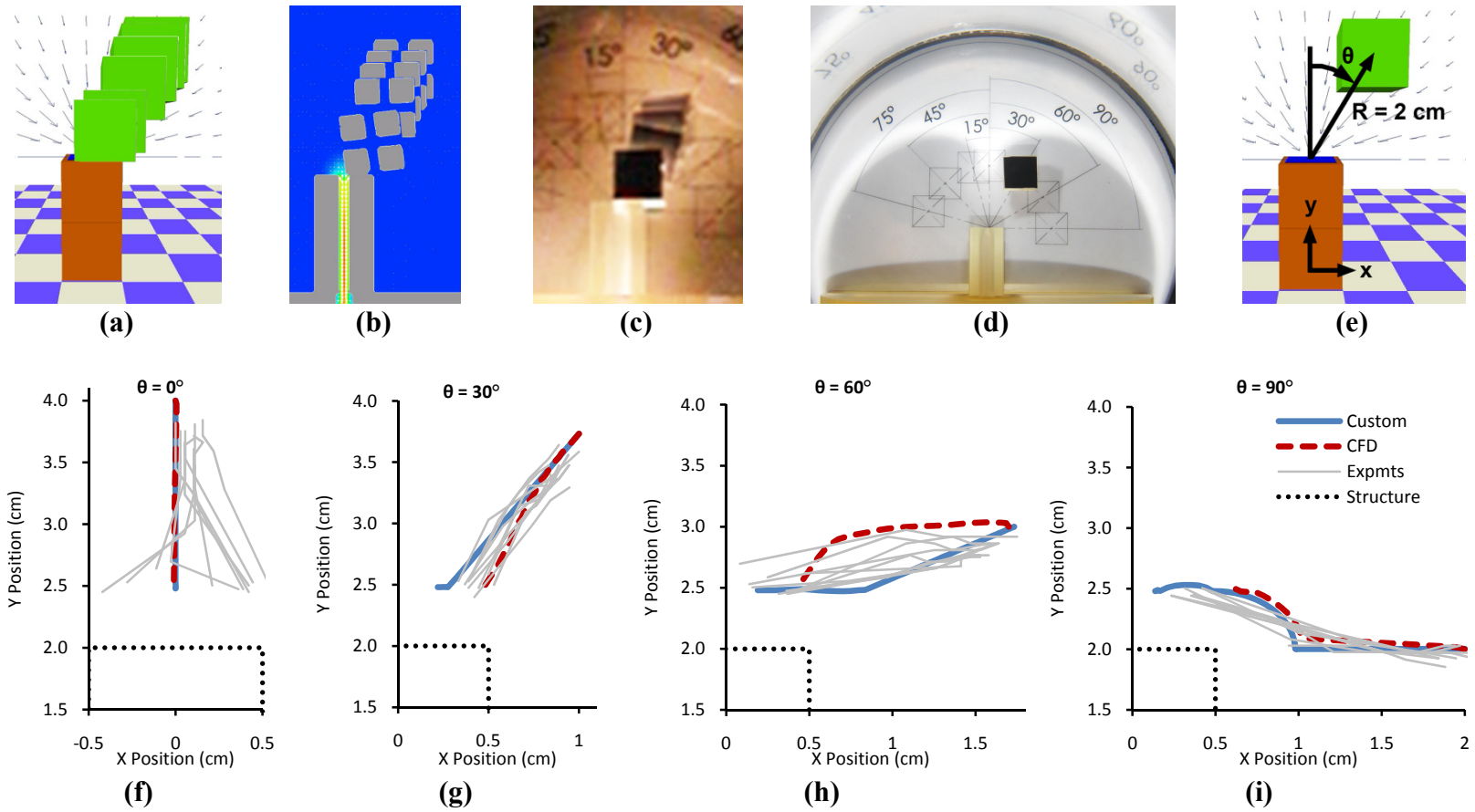


Figure 5.4. Attraction model validation. (a) We simulated the motion of a cube being attracted to the top of a partially assembled structure from various approach angles using our custom software. We then compared the cube paths with those simulated using the more accurate (yet computationally intensive) (b) FLOW-3D commercial software, and [(c) and (d)] experiments. [(f)–(i)] Plots compare the paths taken by the cubes in simulation and experiment for four approach angles θ, as defined in (e).

Figure 5.4(f)–(i) plots the cube motions from the simulations and experiments. In general, there is a very good agreement between the three sets of paths. The biggest discrepancy between the CFD and custom simulations occurs in the $\theta = 30°$ case, where the CFD's hydrodynamic forces cause the cube to move first in the x-direction toward the sink, and then in the y-direction, while the custom-simulation cube moves directly toward the sink. However, both behaviors were seen in experiment.

One feature of the experiments that did not show up in the simulations is that the cubes never aligned directly with the sink, even in the $\theta = 0°$ case. Despite the absence of any near-field alignment force, the cubes in simulation often came to rest near an aligned position, especially when approaching from directly above. However, the experimental cube paths can be seen to bifurcate toward one of the corners of the structure and, hence, always approaching the sink edge first. This demonstrates the importance of some sort of near-field alignment force if cubes are to be assembled on a regular lattice. We discuss potential near-field forces further in the *Near-Field Force* section below.

Overall, the general agreement of the CFD and experimental results with those of our custom simulator for the test case gives us confidence in our fluidic attraction model. In the next section, we use further experiments to validate our custom simulator's assembly rates in a more complex situation.

Experimental Validation of Assembly Rates

In this section, we compare the assembly rates predicted by our custom simulator with those observed experimentally using a test chamber (see Figure 5.5), over a range of sink flow velocities. A single sink on top of a one-cube structure at the bottom of the chamber attracts the cubes that are initially in random positions. In the experimental system, a fluid jet and two sinks on the top of the chamber provide

agitation. In simulation, Gaussian-distributed random agitation forces are applied to the cubes at each time step. The simulation parameters were set as indicated in Table 5.1, except that the sink flow velocity was varied over the range from 280 to 560 cm/s.

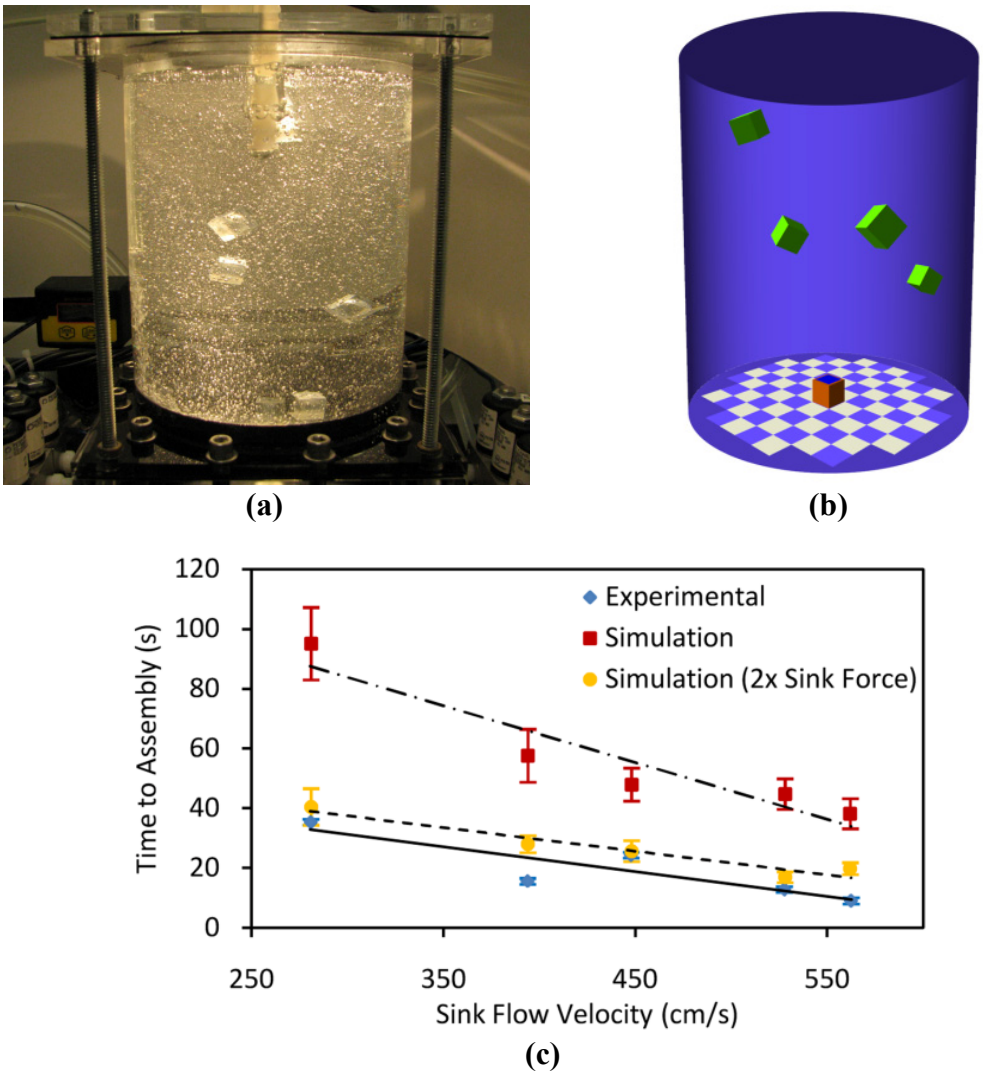


Figure 5.5. Comparison of experimental and simulated test results. Images of the (a) experimental and (b) simulated test tank. (c) Time to assemble a cube on top of a seed cube in simulation and experiment for various sink flow velocities (n = 40). The results of a simulation with a sink force of twice that predicted by our model correlate very well with the experimental results.

The motions of the cubes in the experiment and simulation were found to be qualitatively similar (see video). In each case, the time required for a cube to become attracted to the sink (i.e., time to assembly) was recorded over 40 trials [see Figure 5.5(c)]. The variation in the assembly times due to the sink flow rate was found to be very similar in experiment and in simulation. However, the simulations took an average of about three times as long to assemble a cube. Interestingly, increasing the magnitude of the sink force by a factor of two led to assembly rates that were much more similar to those found in the experiment. This result—like those of the previous section—suggests that our fluid forces model may be under-predicting the hydrodynamic forces applied to a cube by a sink.

Table 5.2 System-Parameter Recommendations

Parameter	Unit	Recommended Value
Agitation (Average Cube Kinetic Energy)	millijoules per cube (mJ/cube)	0.0004
Sink Flow Velocity	centimeters per second (cm/s)	700
Cube Concentration	percentage volume (%)	4
Friction	unitless	0.50
Near-field Force	millinewtons (mN)	$7x10^{-4}$
Cube Mass Centre Offset	millimeter	1.5

System Parameters

One of the two main goals of developing a custom simulator was to be able to predict the effects of changes in key system parameters on the system's performance.

Determining the ideal parameter settings experimentally could be costly, especially for highly interdependent parameters. We thus identified six key simulation parameters that we would like to set using simulations: agitation strength, cube concentration, sink flow rate, friction, near-field force, and cube weighting. For each case, we varied the parameter in question and measured the resulting assembly rates and completion percentages for the assembly of a test shape. Table 5.2 summarizes our system-parameter recommendations based on the results of these simulations. Our custom simulator was written to accommodate the assembly of arbitrary target structures (Figure 5.6), however, all of the following simulations have been conducted on a constant target shape [the wrench shape seen in Figure 5.6(a),(b)] for consistency.

Agitation and Sink Flow Rate

Fluidic agitation was modeled in our simulations as a Gaussian-distributed random force applied to each cube at each simulation time step. These random forces modeled the stochastic forces that are difficult to predict accurately in a computationally efficient way. The results presented in the *Experimental Validation of Assembly Rates* section above suggest that this forms a reasonable model of the agitation created in experiment. The amount of agitation created in this way was calculated as the mean kinetic energy of each cube under the influence of these agitation forces only (in millijoules per cube). Additionally, using the model from the *Fluid Forces Model* section, we can calculate the forces on the cubes due to active sinks as a function of the sink flow rates. Thus, we have two independent fluid force parameters. In this section, we investigate the interplay between these parameters.

One of the first lessons that we learned in running the simulations was the importance of the stochastic agitation to the overall fluidic assembly approach. First, agitation was required as a transport mechanism to move free cubes to activated

bonding sites. Second, agitation was required to counteract the fluidic forces due to open sinks. In the absence of agitation, all the cubes would collapse on any open sinks, clogging them up, and preventing further assembly. However, while agitation was required for assembly, too much agitation was found to be a destructive force. As the magnitude of the agitation increased, the corresponding structure-assembly rates—and, eventually, completion percentage—decreased.

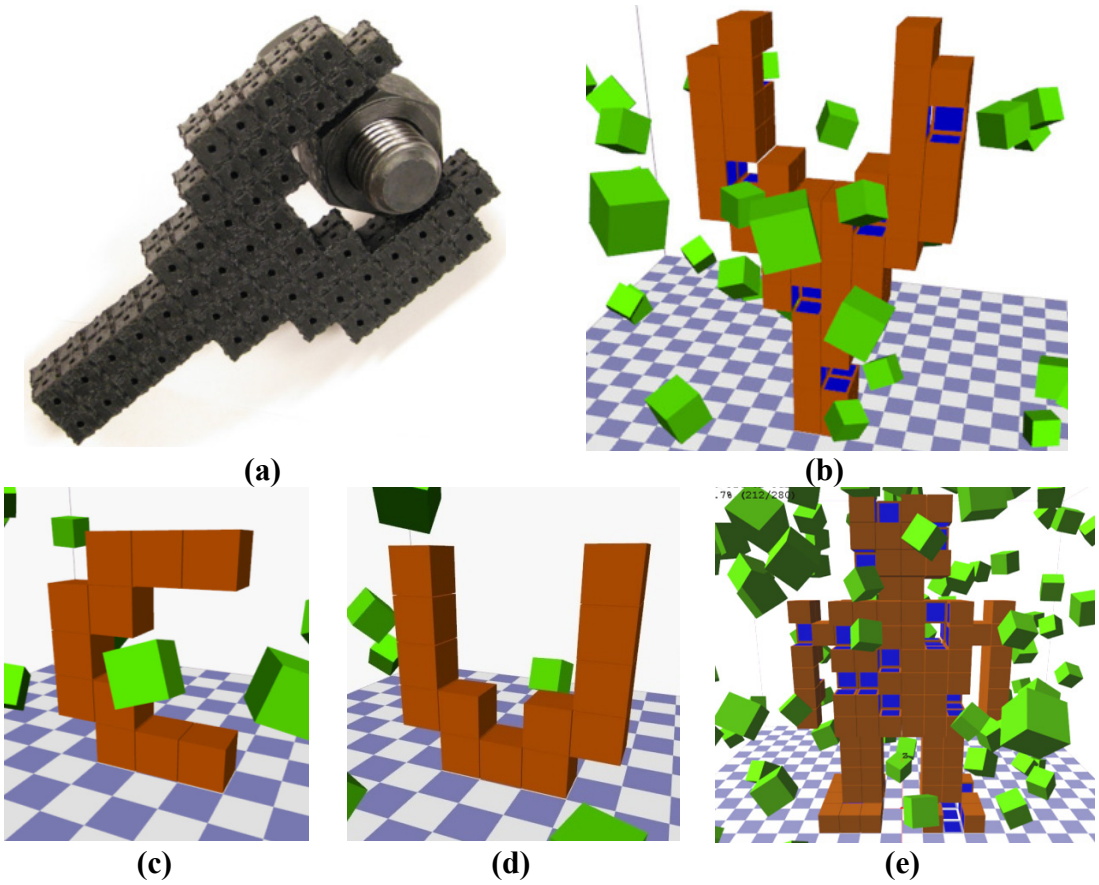


Figure 5.6. Assembly simulator target structures. Our custom simulator was written to accommodate arbitrary target structures. (a) Physical mock-up and (b) simulated assembly of 104-cube wrench target structure. (c) Example C-shaped, (d) U-shaped, and (e) robot target structures.

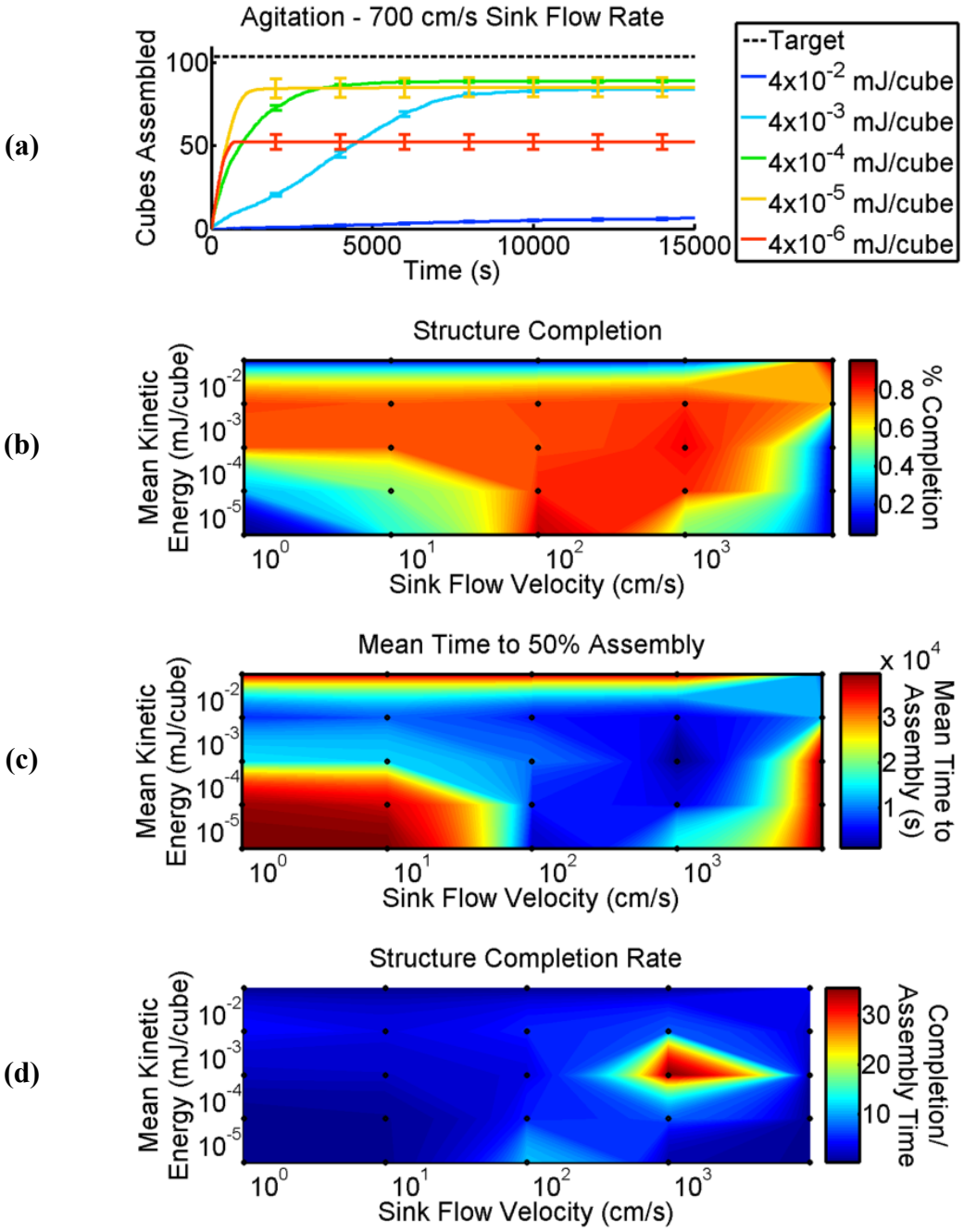


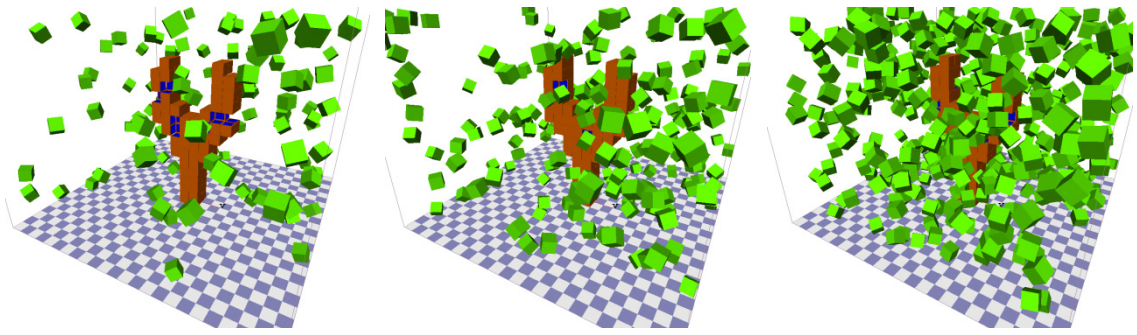
Figure 5.7. Agitation and sink flow rate. (a) Plot of number of cubes assembled versus time for various agitation rates. [(b) and (c)] Contour plots depicting the structure completion and mean time to 50% assembly for various agitation rate–sink-flow rate combinations. (d) Dividing the completion by the mean time to 50% assembly identifies the ideal parameter settings to assemble the most complete structures in the least amount of time.

We quantified these results by running 20 simulations of the wrench test shape at various agitation settings. Cubes were added to the structure following a greedy strategy that attracted cubes to any location within the target structure adjacent to an attached cube. Figure 5.7(a) is a plot of the average number of cubes assembled to the target structure versus simulated seconds over 20 simulation runs. The error bars represent standard error. This plot shows that for a constant sink flow rate (i.e., 700 cm/s), as the amount of kinetic energy imparted on the cubes by agitation increases from 4×10^{-6} to 4×10^{-2} mJ/cube, initially, the assembly completion increases, while the assembly rate decreases. However, after the average kinetic energy increases beyond 4×10^{-4} mJ/cube, both the assembly rate and the completion of the structure decrease. Thus, for the chosen flow rate, there is an optimal amount of agitation that excites the cubes to approximately 4×10^{-4} mJ. In general, it seems as though it is best to use the minimum amount of agitation necessary for cube transportation, thus counteracting the fluid forces attracting the cubes to open sinks. Another interesting trend that is evident in Figure 5.7(b) and (c) is that it is possible to increase the sink flow rate to a velocity of 7000 cm/s as long as the agitation is also increased to 0.04 mJ/cube. In fact, there is a ratio of flow rate to agitation, i.e., 100000 (cm/s)/(mJ/cube) that leads to effective assembly beyond a minimum agitation–flow rate combination of approximately 0.0002 mJ/cube and 20 cm/s. Thus, we found that there was an important relationship between these two parameters.

Cube Concentration

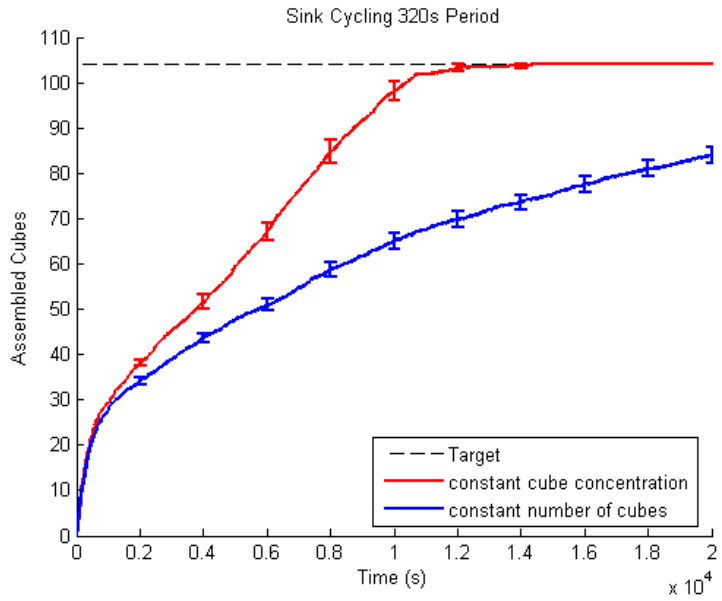
The cube concentration is defined as the fraction of the open assembly tank volume (the total volume minus the volume occupied by assembled structure) that is occupied by free cubes [Figure 5.8(a)]. It may seem natural (and practically more feasible) to keep the total number of cubes in the assembly tank constant over the

course of assembly. However, as cubes assemble onto the structure, the number of free cubes in the tank becomes depleted. This has a negative impact on assembly rates as the probability of an assembly event is proportional to the cube concentration [Figure 5.8(b)]. This suggests that it might be worth the extra effort of maintaining a constant cube concentration in the experimental system.



1% **2%** **4%**

(a)



(b)

Figure 5.8. Cube Concentration. (a) Maintaining a constant cube concentration (as opposed to a constant total number of cubes) was found to be beneficial to assembly. (b) Images from simulations run with cube concentrations of 1%, 2%, and 4% (respectively).

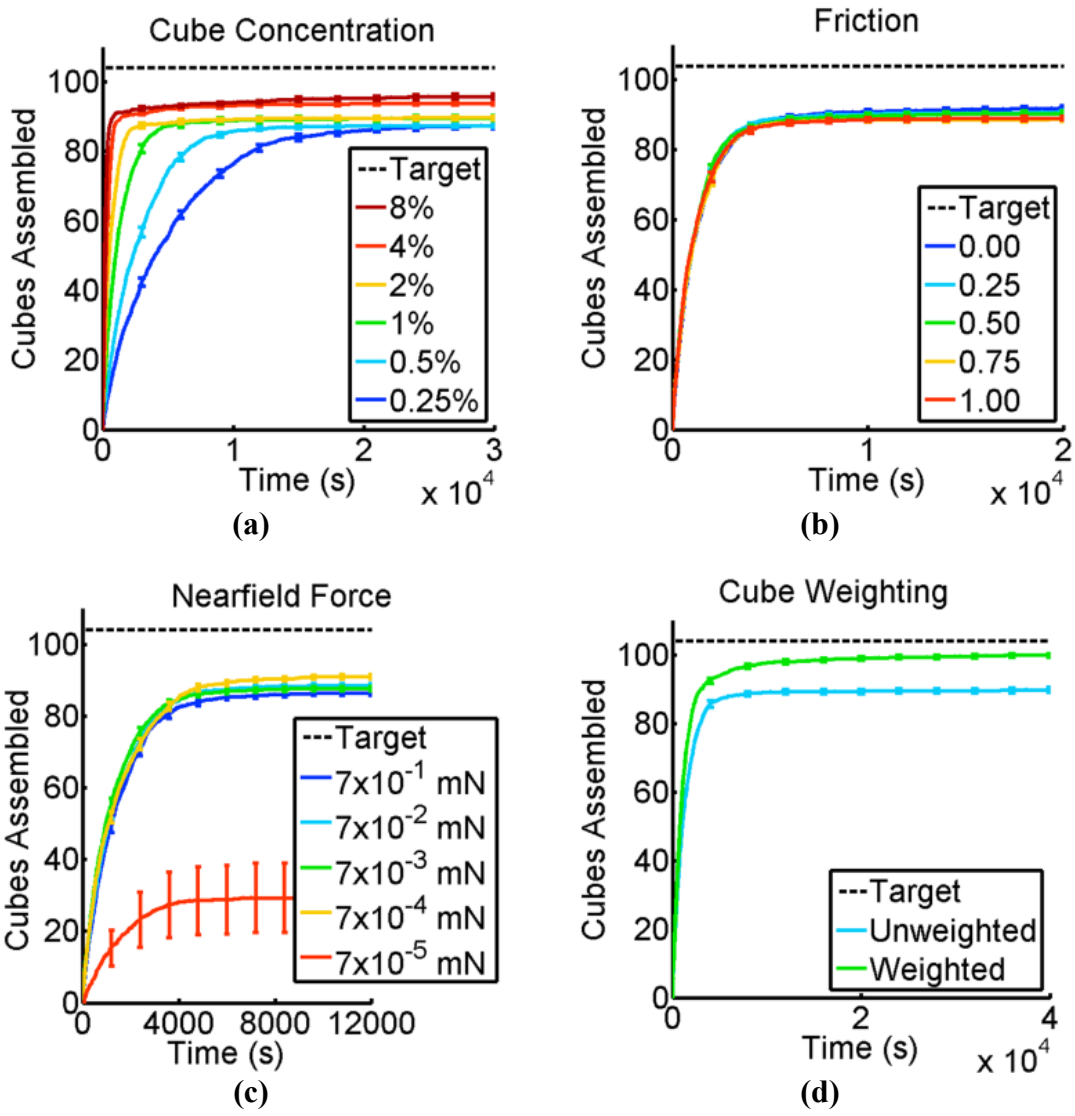


Figure 5.9. System parameters. Plots depicting the number of cubes assembled to the wrench target structure versus simulation time for the simulation-parameter settings. (a) Cube concentration, (b) friction, (c) near-field force, and (d) cube weighting.

Maintaining a constant cube concentration over the course of a simulation, we investigated the effects of changing this concentration in different simulations. The relationship between cube concentration and assembly rates can be seen in Figure 5.9(a), which plots the average wrench assembly curves for various cube concentrations using the optimal agitation and sink flow-rate parameters identified in

the previous section.

The concentration of free cubes was kept constant throughout the simulations by replacing cubes that become attached to the structure with new free cubes randomly placed at the top of the tank. As the cube concentration increases from 0.25% to 8%, we see an increase in the assembly rate and the completion percentage. However, these are subject to diminishing returns (i.e., doubling the cube concentration results in marginal increases). Assuming that there are costs associated with increasing the cube concentration in the experiment (e.g., more excess cubes required for a given assembly, difficulty in observing assembly), we recommend a target cube concentration of 4%.

Friction

In our simulations, a common friction parameter was used for the friction between cubes and between the cubes and the assembly tank wall. Changing this value was found to not have a significant impact on the assembly simulation results [see Figure 5.9(b)]. Our results indicate a slight improvement with lower friction values, most likely because this helped cubes to fit into tight spaces. However, there will probably be lower limits to the friction values achievable in experiment. In the remainder of our experiments, we chose to use a near-worst-case value of 0.95.

Near-Field Force

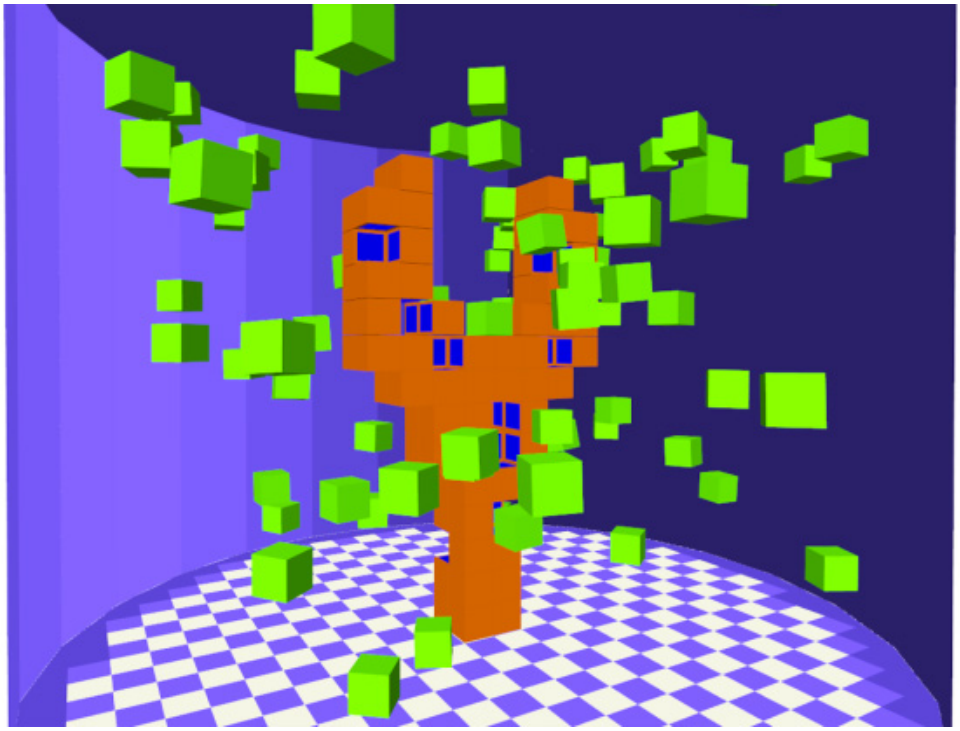
In our simulations, we assumed the existence of a near-field force to attract and align cubes once they approached within a threshold proximity of a sink. This force represents a close range force that acts in conjunction with the sink force but does not act beyond the nearest cube. In previous work, we have investigated the use of permanent magnets and electromagnets as a near-field force for stochastic assembly (White, Kopanski, & Lipson, 2004; White, Zykov, Bongard, & Lipson, 2005). Other

researchers have also investigated the use of magnets (Griffith, Goldwater, & Jacobson, 2005; Klavins E. , 2007), capillary forces (Whitesides & Grzybowski, 2002; Srinivasan, Liepmann, & Howe, 2001), and intermolecular interactions (Winfree, Liu, Wenzler, & Seeman, 1998). Also related is the latching force between modules, which could either be the same as the near-field force or another additional mechanism. For example, in Chapter 2 we conducted studies on the use of passive compliant latches for the assembly of microscale components (Tolley, Baisch, Krishnan, Erickson, & Lipson, 2008). We have also previously investigated the use of active latches for the assembly of 10-cm scaled components (Zykov & Lipson, 2007).

In our simulations, this force was applied to a cube when it approached within 0.8 cube lengths of an attracting cube or growth substrate face. The effect was modeled by a constant force applied to each of the four corners of the cube's closest face in the direction of the corresponding (closest) attracting face corner. While the physical implementation of near-field and/or latching forces is a key challenge for modular robotics systems, further discussion of this topic, as well as an experimental validation this model, is beyond the scope of this study.

Structure-assembly rates for simulations using the assumed near-field force with various force constant values are plotted in Figure 5.9(c). We found that this force had a critical value of 0.0007 mN, below which, very little assembly occurred (i.e., the force was too weak to lock cubes once they came close). However, all the values at and above 0.0007 mN were able to attach cubes to the structure and showed similar results in terms of assembly rate and structure completion. Surprisingly, above the critical near-field force value, increasing the nearfield force actually resulted in a slight decrease in structure completion, perhaps because cubes attached more readily to extremities of the structure, thereby increasing the likelihood of leaving unfilled holes toward the middle of the structure.

(a)



(b)

Figure 5.10. Weighted cubes. (a) Frames taken from experimental testing and (b) image from a simulation of cubes with an inhomogeneous density designed to maintain a single orientation to improve assembly.

Cube Weighting

Weighted cubes have an inhomogeneous density such that the opposing forces of gravity and buoyancy align their bottoms with the horizontal plane (see Figure 5.10). The idea is that maintaining the same orientation will improve alignment with sinks. Cube weighting was tested in simulation by applying the gravity force to the cubes 1.5 mm lower than the volumetric centre (which represents a change in the mass centre).

The curves of Figure 5.9(d) show assembly rates for weighted versus uniform density cubes. We found that weighted cubes assembled into a wrench more quickly and completely than their homogeneous counterparts. In an experimental system, cube weighting has the additional benefit of predetermining the top and bottom faces, which could allow the designer to reduce the complexity of the mechanical and electrical interfaces on those cube faces.

Discussion

The results presented in this chapter highlight the benefits and challenges of simulating stochastic modular robotic assembly. The comparisons of the *CFD and Experimental Validation of Module Attraction* and *Experimental Validation of Assembly Rates* sections suggest that it is possible to conduct such simulations in a computationally efficient manner while maintaining fidelity in module motion during attraction and in the overall assembly rates. However, the assumed models for certain system parameters, such as the near-field and agitation forces, remain to be validated experimentally.

One challenge that was encountered during development of the simulator was the occurrence of non-physical results when a large number of cubes collapsed onto one location. Eventually, numerical inaccuracies or a cube snapping into place would cause all nearby cubes around to interpenetrate. The ODE rigid body solver would then apply abrupt forces in an attempt to eliminate these interpenetrations. The result would be an explosion of cubes. While non-physical, this turned out to be beneficial to assembly as it cleared the way for further assembly. These artifacts can be identified in the simulation data large increases, followed by abrupt drops, in the number of collisions (represented by a blue line on the plot on Figure 5.11). In order to penalize these conditions, we assumed that assembly would have stopped at this point.

Additionally, minimizing abrupt changes in cube positions when they attach to a structure helped alleviate this problem.

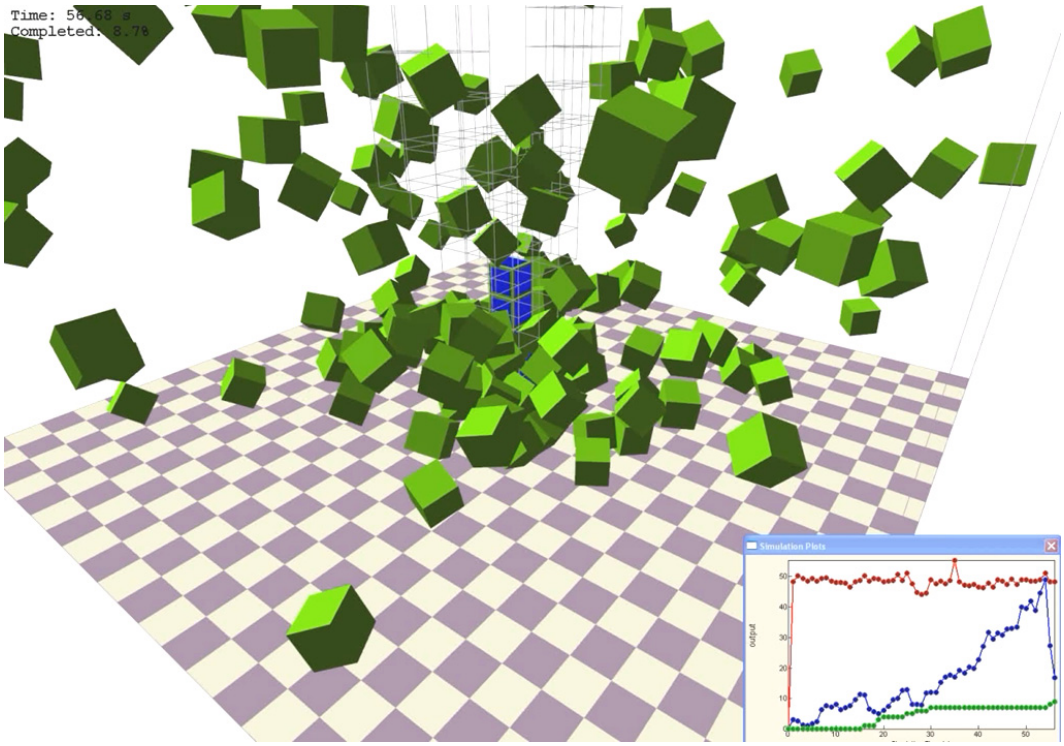


Figure 5.11. Explosive Interpenetrations. A large number of cubes being compacted into a small space can cause excessive cube interpenetrations. This led to explosions of cubes in simulation, especially when coupled with cubes snapping abruptly into place. These explosions were identified by large increases–followed by abrupt drops–in the number of system contacts.

Conclusions

In this chapter, we have presented a computationally efficient simulator to model the stochastic fluidic assembly of robotic modules. We have validated this simulator by comparing its results against those of CFD simulations and a test experimental system. We then used this simulator to study the effects of various system parameters on the speed and accuracy of assembly of a target structure. The results of these

simulations suggest ideal values for design parameters of an experimental fluidic assembly system. We hope that these results will help guide future physical implementations by identifying the importance of the various parameters. As new experimental results become available, these will then be used to further refine our simulator.

In general, the deterministic assembly of modular robots will become increasingly difficult as the size of the modules decreases and the number of components in a target system increases. Thus, we expect that design choices and assembly strategies based on efficient simulations, such as those presented here, will become increasingly important for scalable approaches to stochastic fluidic assembly. In Chapter 6, we discuss a fundamental set of assembly strategies developed using the simulator presented in this chapter for achieving stochastic fluidic assembly. In Chapter 7, we further present an on-line assembly algorithm to select assembly paths that maximize assembly rates throughout the assembly process.

PART IV: CONTROL ALGORITHMS

CHAPTER 6: STRATEGIES FOR STOCHASTIC FLUIDIC ASSEMBLY

Abstract

In this chapter we describe a set of control strategies for stochastic fluidic assembly. Through a set of stochastic fluidic assembly simulations, we discern the strengths and weaknesses of five different assembly strategies and measure their relative performance using the metrics of structure completion and assembly time. Additionally we test these strategies on two topologically different target structures in order to get an estimate of the generalizability of these results.

Introduction

In Chapter 5 we presented a stochastic fluidic assembly simulator to predict assembly rates and qualities for various target shapes and tunable parameters. In this chapter we describe the use of this simulator to develop a set of control strategies for stochastic fluidic assembly. We further discuss the tradeoffs between the various assembly strategies and the ramifications of these results for the design of achievable target structures.

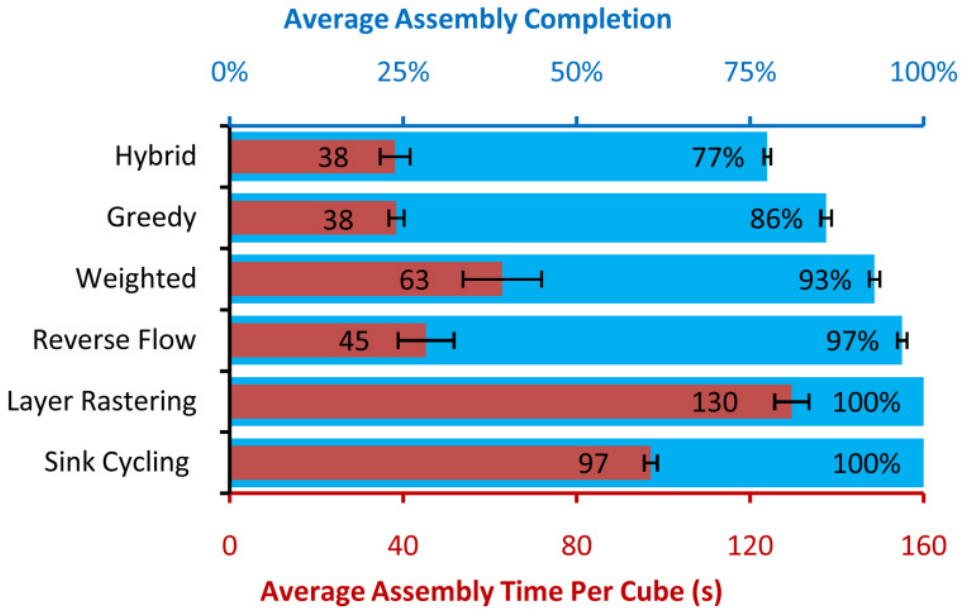
A review of previous modular robotic systems and specifically stochastic modular robotic systems is given in Chapter 5 and will not be repeated here. However, here we consider a few details specific to the control of stochastic modular robotic systems. For a more detailed review of stochastic assembly in general, please see the introduction of Chapter 7.

The type of control required for a modular robotic system depends heavily on its

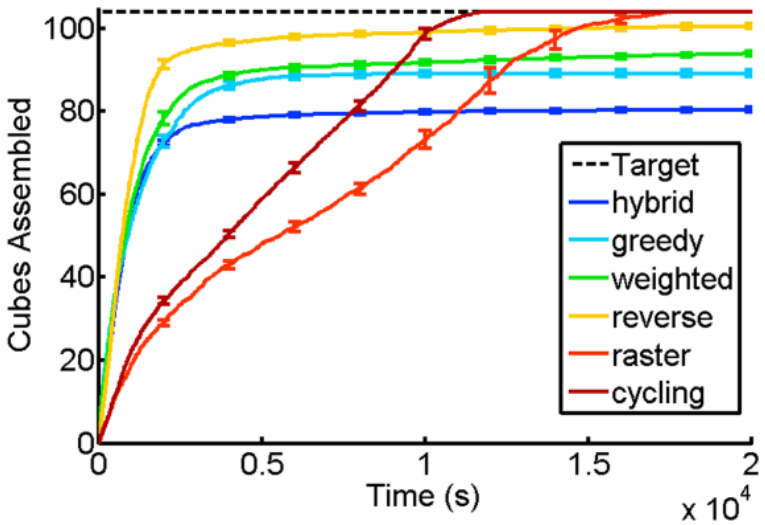
architecture. Many of the systems with mobile modules assemble and reconfigure themselves with both low (module)-level and high (system)-level control (Pamecha, Ebert-Uphoff, & Chirikjian, 1997; Lee & Sanderson, 2002). Stochastic assembly systems avoid the requirements for complex motion-planning control at the module level and instead require only a decision of whether or not to connect two components when they come into contact. This decision may either be made based on local information (which is similar to cellular automata) or made centrally and distributed via intercomponent communication. However, since the arrival time of a component at any given location cannot be predicted deterministically, robust assembly strategies must be employed to account for these uncertainties and accelerate assembly/reconfiguration. Thus, in addition to simplifying module design, a stochastic assembly approach simplifies module-level control requirements, at the cost of increased uncertainty that must be compensated for in the system-level control scheme.

Assembly Strategies

In this section, we present five different strategies for the assembly of arbitrary structures. While it is possible to analyze the admissible assembly sequences for a given configuration to generate assembly paths and/or rules (see Chapter 7 for our work using this approach, as well as a review of related work), this is not the goal of this study. In this initial look at the simulation of 3-D stochastic assembly, our aim is to develop simple strategies that maximize assembly rates while requiring little or no additional system or module capabilities. This is consistent with the overall motivation for stochastic assembly, which is to minimize component functionality. Thus, we have attempted to identify five assembly strategies that require a minimum sensing, actuation, and computation.

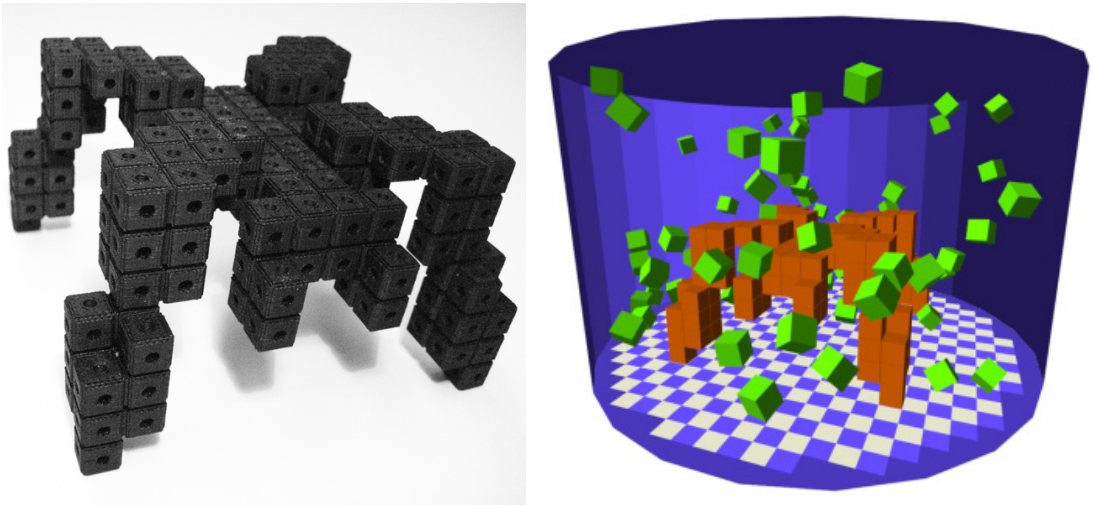


(a)



(b)

Figure 6.1. Comparison of assembly strategies. (a) Summary of assembly completion and time-per-cube statistics and (b) plot of average cubes assembled versus time for the best cases of various assembly strategies. Whereas the *layer-rastering* and *sink-cycling* strategies were found to result in the most complete structures, they also had the highest average time per cube assembled. The *greedy* and *reverse-flow* strategies were much faster but resulted in less-complete structures.



(c) (d)

Figure 6.2. Legged-robot target shape. (a) Physical mock-up and (b) simulated assembly of 174-cube legged-robot structure.

These five approaches that we investigate are 1) greedy assembly, 2) reverse flow, 3) raster filling, 4) sink cycling, and 5) a hybrid raster/greedy approach. These five strategies are described in this section along with the resulting simulation assembly statistics. Figure 6.1 summarizes, for each case, the average assembly completion and the average assembly time per cube. The latter is defined as the average time taken, per cube, to assemble the first 95% of the final structure. (This definition was chosen since assembly typically approached to the final values asymptotically over time, obscuring the time required to assemble the majority of the completed structure.) Note that we have included the results of the greedy strategy with cube weighting in Figure 6.1 due to the significant effect of this parameter on the results of the greedy approach.

We apply these strategies to two topologically different target shapes. In addition to the wrench shape seen in Chapter 5 [Figure 5.6(a),(b)], we also apply these strategies to a legged robot shape (Figure 6.2) in order to get an idea of their general applicability. To be able to compare the results of the various assembly strategies, they have been run with constant parameter settings (700 cm/s flow rate, 0.0004 mJ/cube

agitation, 1% cube concentration, a friction parameter of 0.95, 7×10^{-4} mN near-field forces, and no cube weighting). Two exceptions occur with the reverse-flow and sink-cycling strategies, which were designed to operate with lower amounts of agitation (4×10^{-5} mJ/cube).

Note that the goal of this section is not to provide an exhaustive list of all possible stochastic assembly approaches (indeed, one could imagine a wide range of possible strategies). It is instead meant to explore some of the possibilities and provide direction for future experimental and simulation work.

Greedy Approach

All of the assembly simulations that we have seen so far have followed the same greedy strategy, which can be summed up as "*always open a sink when possible where a cube is required*." This strategy has the advantage of being simple (both algorithmically and in terms of control) and results in quick assembly rates. The plots in Figure 6.1 include separate results for this greedy strategy applied to modules with homogeneous density (which are labeled "Greedy") and weighted modules (which are labeled "Weighted").

The major drawback of the greedy strategy is that it tends to leave unreachable holes, thereby resulting in porous structures [see Figure 5.6(b),(e)]. While it may be possible to adjust a target structure's design to be able to compensate for such defects, or to correct for errors after they have occurred, it would be preferable to avoid them in the first place using an appropriate strategy. This is the goal of the remaining strategies described in this section.

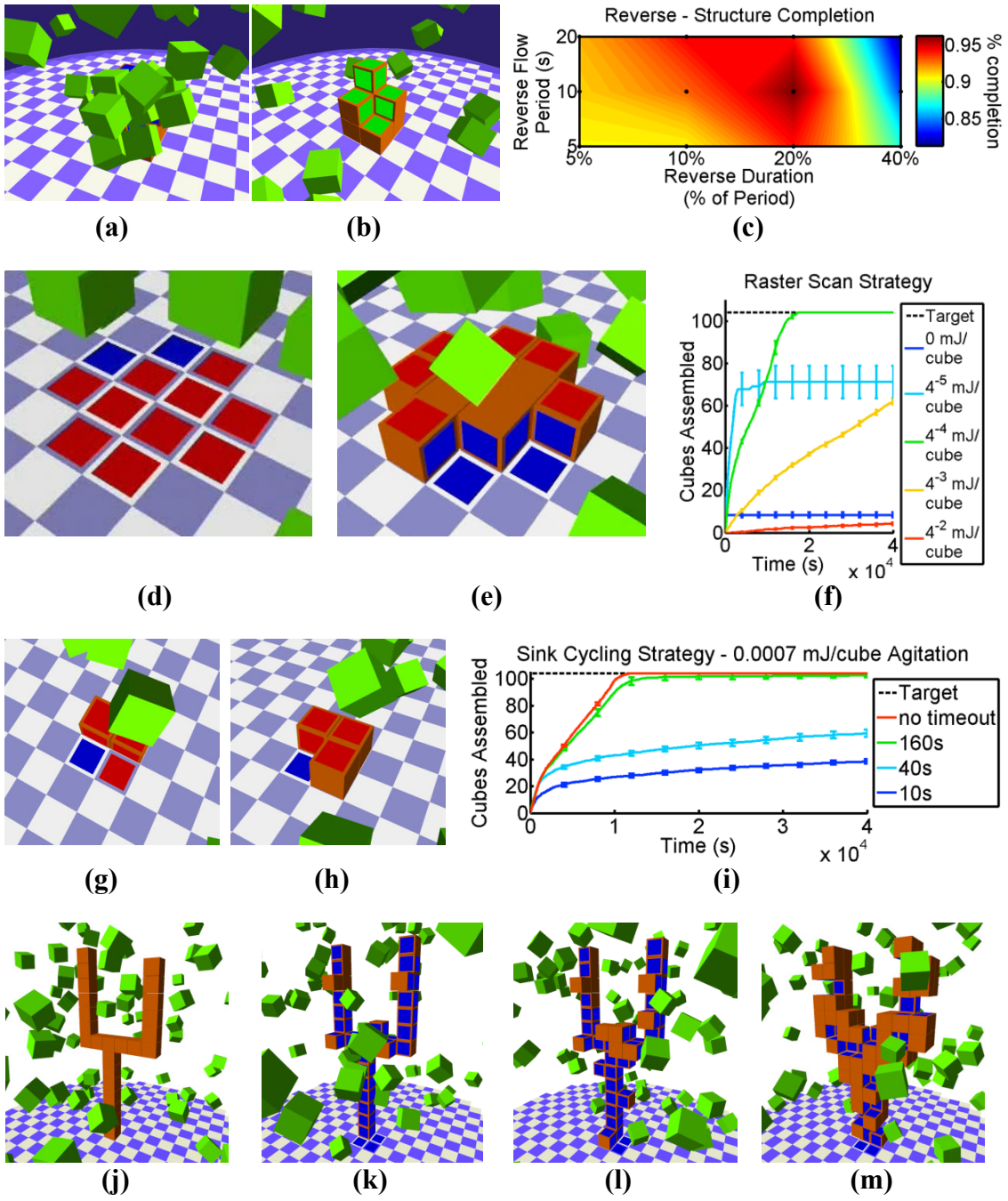


Figure 6.3. Assembly strategy simulations. (a)–(c) Reverse-flow strategy reverses the sink flow periodically to prevent cubes from clumping. (d)–(f) Layer-rastering strategy fills layers from top-left to bottom-right. (g)–(i) Sinkcycling strategy opens only one sink at a time and cycles periodically. (j)–(m) Hybrid greedy/raster strategy guarantees structural strength by first forming a skeletal structure and increases assembly speed by filling in the remainder of the structure quickly.

Reverse Flow

The reverse-flow strategy involves periodically reversing the fluid flow such that the sinks become sources [which are indicated by green squares on the locked red cubes in Figure 6.3(a) and (b)]. The reverse-flow strategy avoids the problem of cubes clumping around the sinks in high sink flow rate or low -agitation conditions. Sources are achieved in simulation by applying the negative of the sink force ((5.11). The two parameters of this approach are the period of the entire ON–OFF cycle and the duration of the reverse portion of this cycle. We varied the cycle period from 5 to 20 s and the reverse duration from 5% to 40 % of this period. Figure 6.3(c) shows the results of this sweep. We found a 10-s period with a 20% reverse-flow duration to be optimal.

Layer Rastering

In order to avoid the "holes" in the structure that plague the greedy strategy, the layer rastering strategy fills in one layer at a time beginning at the top-left and working its way down to the bottom-right [see Figure 6.3(d) and (e)]. While this approach can result in perfect assemblies, it also has two weaknesses: First, it is slow because it has to wait for cubes to come repeatedly to the same part of the structure, thereby exhausting the local supply of free cubes. Second, it is prone to clogging as cubes are always being attracted to the same location. Both of these problems can be alleviated with higher agitation [see Figure 6.3(f)]; however—as shown in the *Agitation and Sink Flow Rate* section of Chapter 5—this is detrimental to assembly rates. Despite this shortcoming, raster filling was found to assemble perfect wrench structures with an agitation of 0.0004 mJ/cube, taking an average of 130 s/cube.

Sink Cycling

Of the strategies that we investigated, the most successful in assembling the

wrench target structure was sink cycling. The way this strategy works is that only one sink is active at a time, and the rest are closed [closed sinks are indicated by red squares in Figure 6.3(g) and (h)]. Once a sink has been open for the specified period without attracting a cube, it is closed, and the next sink on the list of sinks is opened. If a sink attracts a cube, the new cube's sinks are all closed and added to the list. The oldest sink on the list is then opened.

The sink-cycling strategy showed very promising results [Figure 6.3(i)]. While the assembly rates were much slower than those for strategies with more sinks open at a time (despite the fact that the strength of each sink force is divided by the number of open sinks), the progress was much steadier. Because the oldest sink was constantly getting priority, this meant that assemblies had little or no errors. In fact, as the cycling period was increased to 160 s, most of the simulations resulted in perfect assemblies. Surprisingly, when the cycling period was increased to infinity (i.e., there was no cycling at all), the strategy resulted in perfect assemblies. Thus, it turns out that the automated switching was unnecessary. Given enough time, simply opening one sink until it becomes filled and then opening the next is sufficient to produce a perfect assembly. Of course, this assumes that there are no assembly errors.

Raster/Greedy Hybrid

The weakness of the greedy algorithm is that the random nature of the assembly process means that the locations of holes in the structure are unpredictable. The layer-rastering approach guarantees structure completion by following an ordered assembly pattern, but at the cost of long assembly times. The goal of the raster/greedy hybrid strategy was to assemble a structure more quickly than the pure raster strategy while maintaining some guarantees about the integrity of the structure. This was accomplished by first assembling a skeletal structure using the deterministic raster

approach and then filling in the remaining structure in a greedy manner [see Figure 6.3(j)–(m)]. Figure 6.1 shows that while the average structure completion was much less than any of the other strategies (77.5%), this strategy was indeed as fast as the greedy approach (38 s/cube). This gives the designer the option of either explicitly or automatically specifying the required skeletal structure and fill regions based on the balance between the required structural strength and the allotted assembly time.

Legged-Robot Target Shape

In order to examine the general applicability of the results of the various assembly strategies developed for the 104-cube wrench target shape, we tested these strategies on a new, topologically different target shape. The chosen target shape is a 174-cube legged robot (Figure 6.2). This shape is fundamentally different in that it seeds from four separate sinks (the four feet), growing four limbs that come together at the body. This poses the new challenge of connecting separate parts together during the assembly. Our goal was to determine how the previously developed assembly strategies would handle this case relative to the wrench case.

The results of this comparison are summarized in Figure 6.4. Figure 6.4(a) compares the average structure completion at the end of 20 runs for the various assembly strategies, whereas Figure 6.4(b) compares the average time taken per cube to assemble the first 95% of the final structure. One of the most significant differences between the two sets of results is the increased effectiveness of the greedy approaches in assembling the robot target shape, which is most likely due to the more skeletal nature of this shape. Its higher surface-area-to-volume ratio of 2.63/L versus 2.04/L for the wrench shape (where L is the cube length) makes it more difficult to leave holes while assembling the robot structure. The reverse-flow strategy has also increased effectiveness, resulting in a near-perfect average assembly completion in

less time. By comparison, the more complicated layer-rastering and sink-cycling strategies have become less effective, thus taking longer per cube than before. Sink cycling also no longer results in perfect assemblies for the legged -robot shape.

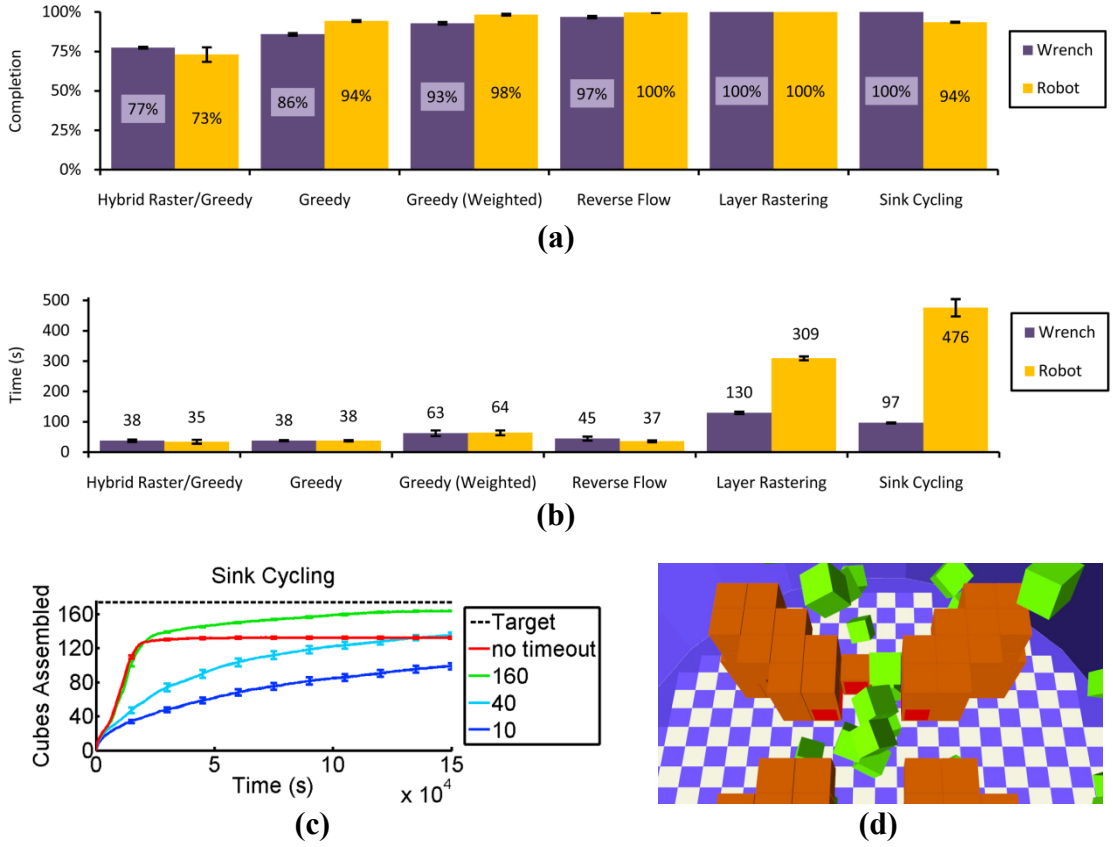


Figure 6.4. Target-shape comparison. (a) Assembly completion and (b) average assembly time per cube for the tested assembly strategies applied to the wrench target shape and the topologically different legged-robot shape. (c) Average number of cubes assembled to the legged robot-shape vs. time using the sink-cycling strategy. (d) Example of failure mode of sink-cycling strategy with no timeout for the legged robot shape.

A comparison of the sink-cycling strategy for the wrench [see Figure 6.3(i)] and legged -robot assembly results [see Figure 6.4(c)] highlights one of the challenges in assembling the latter structure. In the wrench case, cycling sinks with no timeout resulted in perfect assemblies since it was not possible to get into a situation where a

cube could not physically be attracted to a certain location. However, with the more complex legged-robot shape, this approach only worked until it became necessary to insert a cube between two assembled cubes to connect two leg segments [see Figure 6.4(d)]. At this point, assembly halts with an incomplete structure. On the other hand, a cycling period allows assembly to continue (slowly) beyond this point. Thus, this connecting problem is an issue that remains to be addressed.

Scaffolding

Another challenge raised by the legged robot shape is that once the four legs are assembled, they create a sort of barrier that slows down the diffusion of free cubes to the centre of the assembly chamber. This significantly increases the assembly time of the body portion of the shape since free cubes must make their way through and increasingly complex maze of assembled legs to arrive where they are needed. One potential solution to this problem is used frequently in traditional construction: scaffolding. Here, the idea is to assemble extra cubes supporting the body portion of the legged robot shape that are later rejected back into the flow. Thus, the body can be assembled prior to–or in parallel with–the legs, avoiding the diffusion problem mentioned above. However, in addition to requiring more modules for assembly, this approach has the potential to increase assembly times if too many scaffolding cubes are added to the design.

Figure 6.5(a) is an image of the legged robot target shape (dark cubes) with a minimal scaffold consisting of two columns (nine extra cubes) supporting the main body (light cubes). Figure 6.5(b) shows the simulation results for the various assembly strategies applied to the scaffolded shape versus the unscaffolded shape. Surprisingly, in most cases scaffolding was detrimental to assembly in that it either decreased structure completion (Greedy and Sink Cycling), or increased assembly times (Layer

Rastering and Sink Cycling). Only in the reverse strategy case was scaffolding beneficial, reducing assembly times. While this approach shows promise in alleviating some of the problems of assembling complex structures stochastically, more investigation is required to determine how and when it can be applied to greatest effect.

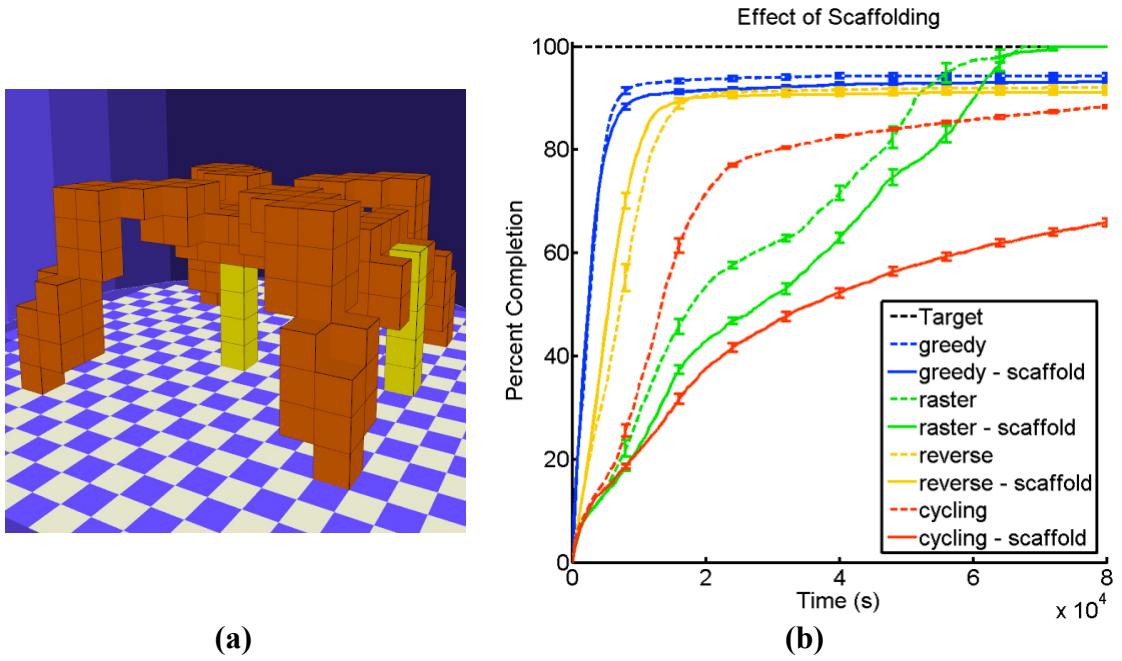


Figure 6.5. Scaffolding approach. (a) Scaffolding support (yellow) allows the legged robot's body to be assembled in parallel with its legs. (b) Comparison of structure completion vs. time for various assembly strategies with and without scaffolding.

Discussion

One of the central challenges of stochastic assembly is the development of strategies that are capable of building a desired target structure while coping with the indeterminate nature of the component supply. The results of the assembly strategy simulations presented here demonstrate the tradeoff between rapid, error-prone

assembly and more deliberate alternatives. It has become clear that while traditional, serial-assembly approaches are possible in such a stochastic environment, more complex parallel approaches have the potential to greatly improve the results. We have presented strategies that take advantage of parallel assembly while guaranteeing at least some part of the structure is error-free.

Conclusion

In this chapter we have discussed the use of the simulator presented in Chapter 5 to develop various strategies for stochastic fluidic assembly. We tested these strategies in simulation by measuring their effects on the speed and accuracy of assembly of two topologically different target structures. The assembly strategies presented here provide a basis for stochastic assembly that—consistent with the aims of this assembly approach—require a minimum of additional system or module capabilities.

The results of the assembly strategies we tested on the two target shapes suggest that as the complexity of the target structure increases, so does the difficulty of achieving error-free assemblies. One approach to dealing with this problem may be to predict the statistical properties of the target structures based on the error rate and design structures that can tolerate an acceptable amount of imperfections. An alternative would be to design some sort of error-correction mechanism could be used in conjunction with a simple assembly approach to achieve complex, error-free structures. A third option would be to devise an on-line assembly algorithm that plans a parallel assembly path and identifies eligible assembly locations that are adjusted on-the-fly in response to new module arrivals. This third option is the approach that we describe in Chapter 7.

CHAPTER 7: ON-LINE ASSEMBLY PLANNING FOR STOCHASTICALLY RECONFIGURABLE SYSTEMS

Abstract

Stochastic assembly approaches reduce the power, computation, and/or actuation demands on assembly systems by taking advantage of probabilistic processes. At the same time, however, they relinquish the predictability of deterministic alternatives. Here we describe an approach to assembly planning for stochastic assembly systems where the spatial and temporal availability of modules is unpredictable, either due to physical uncertainty, resource fluctuations, or large numbers of uncoordinated agents. We propose a parallel assembly algorithm that is guaranteed to find an assembly path for finite-sized, contiguous objects. The path is near-optimal subject to domain-specific local assembly constraints and costs. The algorithm achieves this by sampling the space of possible disassembly paths from the target structure down to the initial state, and sorts them according to certain criteria. Thus, at least one path to a complete target structure is guaranteed at every stage of assembly. The assembly path is computed on-line and thus can adapt to constraint changes during assembly. We demonstrate the application of our algorithm to a number of different target shapes subject to a variety of assembly constraints and costs.

Introduction

The scaling of modular robotics and other reconfigurable systems to large numbers of small-scale units is reaching its limits due to a variety of technical challenges (Yim, et al., 2007). The high power, connectivity and actuation demands of most current approaches to modular robotics approach have led to highly complex modules.

Similarly, the uncertainty associated with manipulating very small elements is challenging for deterministic manipulation approaches. Stochastic assembly processes have the potential to reduce these demands by taking advantage of probabilistic processes for various tasks such as motion planning (Varshavskaya, Kaelbling, & Rus, 2008; Werfel & Nagpal, Three-dimensional construction with mobile robots and modular blocks, 2008; Ayanian, White, Halaz, Yim, & Kumar, 2008), or locomotion(White, Zykov, Bongard, & Lipson, 2005; Griffith, Goldwater, & Jacobson, 2005; Napp, Burden, & Klavins, 2006; Gilpin, Kotay, Rus, & Vasilescu, 2008). In the latter case, modules take advantage of stochastic environmental motions for the transportation of robot components to target locations and/or the removal of unwanted components. This approach reduces the requirement of module mobility at the cost of increased target structure design and assembly planning.

The challenge of assembly planning under uncertainty is relevant to fields beyond modular robotics, to general structure reconfiguration(Nagpal, Shrobe, & Bachrach, 2003; Werfel, Bar-Yam, Rus, & Nagpal, 2006; Klavins, Ghrist, & Lipsky, 2006; Lerman, Jones, Galstyan, & Mataric, 2006; Hjelle & Lipson, 2009). In these situations, a target structure or system is assembled from a set of components whose availability or timing is uncertain, either due to resource fluctuations, unreliability of assembly operations, or other stochastic and distributed control aspects. For example, this situation occurs when multiple robots need to coordinate the joint construction of a structure but the location and state of individual robots is not always known.

In this chapter we present an approach that automatically plans the stochastic assembly of a target structure without knowledge of the times or locations of component availability. Our algorithm takes into account parallel-assembly capabilities, where assembly can be accelerated by promoting assembly at many sites in parallel. The algorithm also guarantees a path to error-free assembly. In order to

accomplish this, our algorithm must determine the next set of locations to allow module attachment, at each stage of assembly. This is achieved by sampling the graph of all possible paths from the current state to the target structure and following those that leave the most options open. For each of these samples, the assembly problem is solved by beginning with the final structure and working backwards, removing one accessible module at a time. Thus each sample is a valid path to an error-free target structure. The potential locations for the next assembly stage are then selected from among those most frequently encountered while sampling. Using this approach, at least one path to a complete final structure is guaranteed at every stage of assembly.

Stochastic Modular Robotic Systems

We distinguish three specific types of stochastic modular robotic systems to which our assembly approach may be applicable (Figure 7.1): Those which form assemblies from modules, with modules and assembled structures moving stochastically in an agitated environment, those with modules in a stochastic fluidic environment depositing onto fixed assemblies, and those with decentralized robots depositing structural elements. In the first case modules move about stochastically due to simulated Brownian motion and decide whether or not latch together when they collide into one another. This concept has been demonstrated experimentally in two dimensions by floating robotic modules on an air table which is shaken to create stochastic motion (White, Kopanski, & Lipson, 2004; Griffith, Goldwater, & Jacobson, 2005; Napp, Burden, & Klavins, 2006). These modules have electromagnets, actuated permanent magnets, or actuated latches on each face that are used to selectively bond modules together. The assembly and reconfiguration demonstrated by White et al. (2004) followed a hand-coded assembly plan. The modules of Griffith et al. (2005) ran hand-coded finite-state machines in order to

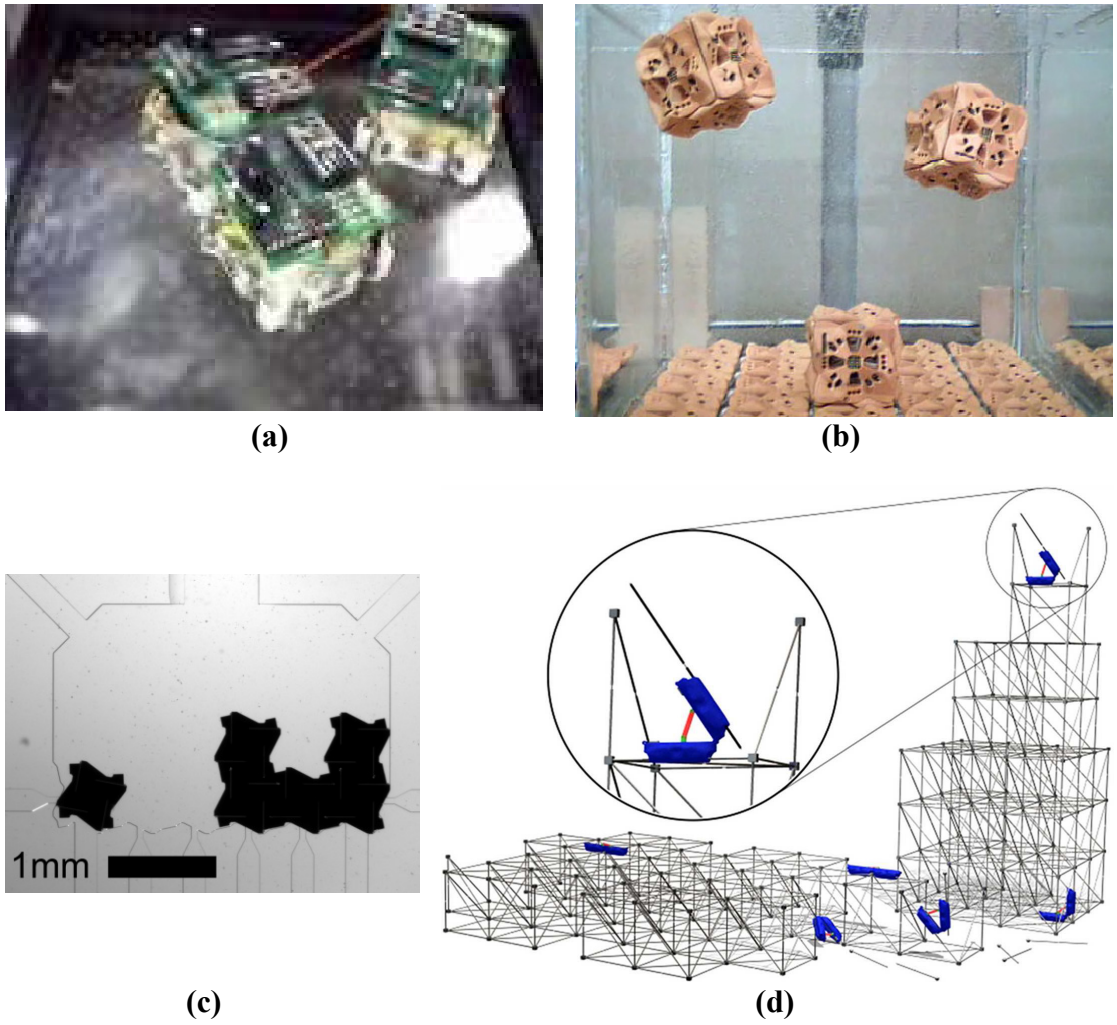


Figure 7.1. Stochastic robotic assembly systems. Various types of stochastic assembly approaches have been proposed for robotic assembly. In other work, we have investigated systems that assemble stochastically (a) on an air table with agitation(White, Kopanski, & Lipson, 2004), (b) in a fluidic tank(Zykov & Lipson, 2007), (c) on a microfluidic chip [see Chapter 1, (Tolley, Krishnan, Erickson, & Lipson, 2008)], and (d) from structural elements manipulated by many simple truss-climbing robots (Lobo, Hjelle, & Lipson, 2009). Taking inspiration from nature, these approaches rely on stochastic processes for component transportation and/or control. The present work investigates an assembly algorithm for systems such as these that can compensate for the unpredictable availability of components.

reproduce linear strings of modules, or to produce repeating linear or planar patterns. Napp et al. (2006) define graph grammars that are used to direct the assembly of their Programmable Parts into target structures. Various assembly and disassembly steps are treated like concurrent chemical reactions, the rates of which can be tuned to achieve a desired steady state.

A second type of system uses stochastic fluid agitation to transport modules to where they are selectively assembled to a fixed structure [Figure 7.1(b), (Zykov & Lipson, 2007; White, Kopanski, & Lipson, 2004; Krishnan, Tolley, Lipson, & Erickson, Hydrodynamically tunable affinities for fluidic assembly, 2009)]. In this case, the selective bonding between modules is achieved using permanent or electro-magnets, latches, and/or hydrodynamic forces. These efforts have demonstrated the assembly of a small number of modules on length scales spanning 500 μm to 13 cm. However, automated assembly algorithms for these systems have not been presented. A detailed description of this stochastic fluidic assembly approach is given in Section 5.

The third type of relevant stochastic modular robotic system involves passive structural modules which are deposited by active modules [(Terada & Murata, 2004; Jones & Mataric, 2004; Everist, Mogharei, Suri, Ranasignhe, Will, & Shen, 2004; Detweiler, Vona, Yoon, Yun, & Rus, 2007; Hjelle & Lipson, 2009), Figure 7.1(c)]. Werfel and Nagpal (2008) describe an assembly approach for such a system in which cubic structural modules are assembled by active modules. Their approach is to have the structural modules determine acceptable adjacent locations for new modules which are transported by mobile robots. This is done by defining a set of rules for selecting block deposition locations that avoids leaving inaccessible gaps in the final structure. The robots then find locations to deposit their blocks by communicating with the structural modules as they climb over the structure. In order to minimize

communication and make the robots as simple as possible, one control scheme they propose is to have the robots move about randomly until a suitable assembly location is encountered. However, even deterministic robot control schemes would most likely lead to unpredictable module arrival times and locations due to their decentralized nature.

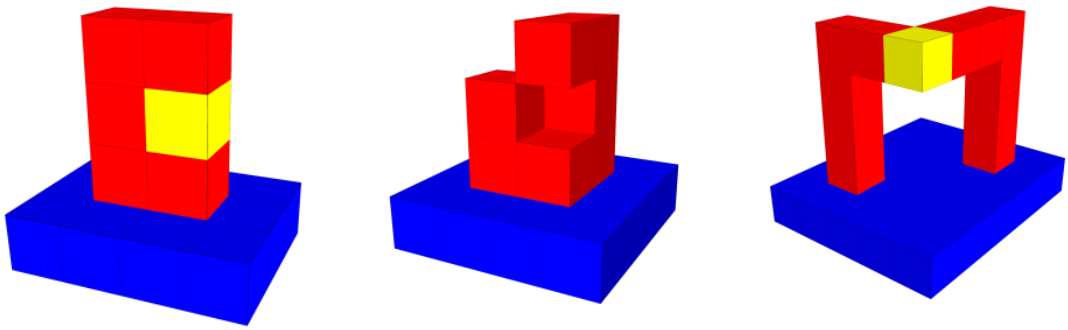
In fact, even centrally controlled systems that are able to deterministically coordinate assembly with a small number of robots will most likely be forced to revert to stochastic decentralized control when they scale up to large numbers of interacting robots. Thus, we see that there are a wide range of proposed and experimental robotic systems that require assembly planners capable of coping with stochastic components availability.

Problem Statement and Assumptions

Our aim is to develop a general, high-level planning algorithm applicable to any of the types of stochastic modular robotic assembly systems described in the previous section. The relevant common aspect of all three systems is that the unknown temporal and spatial availability of the assembly modules. The goal is to be able to input a target configuration (i.e. shape map) and have the planning algorithm autonomously identify a path to error-free assembly. Due to unpredictable component availability, multiple potential assembly paths are to be pursued in parallel. Thus, the algorithm must be able identify a set of valid locations to which modules can be attached at any given stage of assembly. We assume kinematic constraints on assembly based on the local geometry of the assembly site (e.g. situations where a module could be assembled but does not have access). We also assume that there is some cost associated with every assembly event that varies from site to site (e.g. assembly takes longer based on the local geometry). Additionally, we consider the case where there

are practical limitations on the number of locations that can be actively promoting assembly at any given point in time. Thus the goal is to choose the subset of the valid locations at each stage of assembly in a way that reduces the overall assembly cost.

There are a number of greedy assembly planning strategies, including trivial strategies such as bottom-up raster-scanning or randomly choosing the next most accessible unit by some local criteria. Such strategies work well in some conditions but may result in errors or fail to complete assembly for different system parameters or target structures. In Chapter 6, we provided a review of a number of such strategies (Tolley, Kalontarov, Neubert, Erickson, & Lipson, 2010). Our goal here is to find an algorithm that is guaranteed to find a valid assembly sequence, and one that is near-optimal subject to a specified cost function.



(a) **(b)** **(c)**

Figure 7.2. Example configurations. (a) The highlighted module could not be added last since it is impossible to insert a module directly between two others (above and below, in this case). Thus, the highlighted module could not be removed first by Algorithm 1. (b) It is impossible for a module to be inserted into a location with four existing adjacent modules. (c) The worst-case situation for a breadth-first disconnection check function is a module – such as the highlighted one seen here – that connects two long protrusions.

While we believe this algorithm could be useful for systems with various component morphologies—such as the truss assembly robots seen in Figure 7.1(c) (Hjelle & Lipson, 2009)—we limit the discussion here to cubic modules that assemble

on a rectilinear lattice for convenience and due to the popularity of this morphology in modular robotics. Thus, the main kinematic constraint is the inability of the system to insert a module directly between two opposing modules [Figure 7.2 (a)]. Assembly occurs on a flat plane to which initial modules attach. All modules must be connected to the ground plane either directly or indirectly through other modules at all times during assembly. Modules are assumed to be able to bond adjacent faces together when assembled and the loads on these bonds are not considered during assembly. While mechanical considerations such as bond strength may be irrelevant for weightless (e.g. marine or space) applications, they could also be included in the assembly cost calculation for terrestrial applications. We also do not consider the possibility of removing modules during assembly (e.g. for scaffolding).

Stochastic Assembly Algorithm

Assembly Sequence Generation

One approach to achieve the goals described in the previous section is to establish a set of conditions that can be checked to verify if the addition of a module at a given location will eventually result in an unfillable gap in the structure (Werfel & Nagpal, 2008). However, there are two drawbacks with this approach: First, it is difficult to enumerate a simple set of conditions that can be proven to guarantee the assembly of a target structure without placing restrictive limitations on the shape of the target structure (such as having no loops or holes in any plane). Second, for the case where only a limited number of assembly sites can be activated in parallel, this approach gives no indication of how desirable any particular valid location is with respect to any others. While a certain location may be valid in that it does not prevent successful assembly, it may constrain further assembly in an undesirable way.

In order to address these drawbacks we look at the problem from another direction. This approach is inspired by the experience of how much easier it is to take a puzzle apart than to put it back together. Instead of solving the hard problem of determining where to *add* modules without leading to problems further down the line, we start with the completed structure and solve the easier problem of finding modules to *remove*. As long as the constraints on addition, such as not inserting a cube directly between two others, are reversed during removal (cannot remove a cube from directly between two others), simply reversing the sequence of removals gives a valid sequence of additions. This circumvents the problem of determining what complications a particular addition will cause in the future because we already know what the future looks like. We can get an estimation of the expected assembly time for this sequence by adding up the expected time for the reverse of each removal step (calculated based on the local geometry). Note that other disassembly planning algorithms have been described (Gilpin, Kotay, Rus, & Vasilescu, 2008) but those were designed for the explicit purpose of disassembly and therefore did not take into account forward assembly considerations.

The process for obtaining an assembly sequence by disassembling a target structure is described in Algorithm 1. This algorithm takes a target structure to be assembled and returns a valid assembly sequence and the associated expected assembly time. For each iteration of the main *while* loop, the algorithm selects random modules in the structure until it finds one that is both accessible and does not disconnect the structure when removed. A module is defined as accessible if it could have been assembled last subject to the assembly constraints. In the simplest case this only involves checking if there are two modules adjacent on two opposing faces. However, in the case of hollow objects, it may be necessary to also verify that a free path exists between the module in question and the location of the module source.

Algorithm 1 Assembly Sequence Generator

```
initialize structure with user shape map
while modules remain in structure do
while no accessible module found do
 candidate-module ← random module in structure
     if candidate-module is accessible then
       if NOT RemovingModuleDisconnectsStructure (candidate-module, structure)
then
          add candidate-module assembly cost to cost-sum
          add candidate-module location to sequence-list
          remove candidate-module from structure
        end if
      end if
   end while
end while
reverse sequence-list to obtain assembly sequence
```

```
RemovingModuleDisconnectsStructure(module, structure)
num-neighbors ← number of modules adjacent to module in structure
if num-neighbors = 1 then
   return false
end if
neighbor ← random neighbor adjacent to module
structure ← (structure – module)
num-connected-neighbors ← 1
breadth first search through connected modules in structure adjacent to neighbor
if connected neighbor found then
   increment num-connected-neighbors
   if num-connected-neighbors = num-neighbors then
      return false
   end if
end if
return true
```

The disconnection check is an important part of this algorithm. Without it, structures could violate the requirement of being connected at all times during assembly. Our approach to checking connectivity is to remove the module in question and check if its neighbors remain fully connected to other modules or to the ground plane. For the purposes of the disconnection check, we also count the ground plane as

a neighbor. This problem is simplified by two observations:

1. If there is only one neighbor, then there are no parts to be disconnected, thus no check is necessary.

2. If there are more than three neighbors, then this module is not accessible to begin with. This is because it is not possible to choose four cubes neighboring a given cube without two of the chosen cubes being opposite one another [e.g., Figure 7.2(b)]. Thus, by definition, a cube with four or more neighbors is not accessible.

Thus, we need only consider the two- and three-neighbor cases. Our approach is to start with a random neighbor and perform a breadth-first search through connected modules to find the other neighbor(s), again while considering the ground plane as a module with many possible connections. This search ends when either all neighbors are found to be connected (thus no disconnection), or all connected modules have been searched and all neighbors were not found (thus disconnection). A breadth-first search is used since in a majority of cases, the neighbors will be connected within a few steps by nearby cubes. The worst-case scenario is the removal of the only module connecting the ends of long protrusions [e.g., Figure 7.2(c)]. However, in general we expect the disconnection check to take a small number of steps.

In order to prove the effectiveness of Algorithm 1, we require three definitions and two lemmas. First we define an *admissible structure* as a finite-sized, contiguous structure composed of regular cubes assembled on a rectilinear lattice. Second, we define a cube of an admissible structure to be *accessible* if it is not bounded on two opposing sides by neighboring cubes or a boundary. (As discussed above, a cube with no neighbors or boundary on at least three adjacent sides is, by definition, accessible.) Finally, we define a structure cube to be *removable* if it is both accessible, and can be

detached without separating the remaining structure into pieces that are disconnected from each other and the boundary plane.

Lemma 1. *Any admissible structure will have at least one accessible cube, whether or not it is bounded on one side by a rectilinear plane.*

Proof. We prove Lemma 1 by construction. In describing the locations of cubes within an admissible structure, which define the three rectilinear directions x, y, and z such that the coordinate $(i, j, k), i, j, k \in \mathbb{Z}$ corresponds to a cube location a distance of i, j, and k cubes from an arbitrarily chosen origin in the x, y, and z directions, respectively. If the structure is bounded on one side by a plane (either finite, or infinitely sized), we additionally define these directions such that the plane is perpendicular to the z-axis in the negative-z direction [Figure 7.3(a)].

Since admissible structures are finite-sized, any admissible structure must have finite dimensions in the *x*, *y*, and *z* directions. Thus, there exists some integer *Z* corresponding to a *z*-plane that contains at least one cube in the structure, but beyond which there are no more cubes in the positive *z* direction [Figure 7.3(b)]. Likewise, since the structure is finite in the *y*-direction, there exists some integer *Y* corresponding to a row in this *z*-plane that contains at least one structure cube, but beyond which there are no more cubes in the positive *y*-direction in this plane [Figure 7.3(c)]. Finally, in this row, there exists some integer *X* that corresponds to a structure cube beyond which there are no more cubes in the positive *x* direction in this row [Figure 7.3(d)]. Thus, the cube with coordinates (*X*, *Y*, *Z*), with no neighbors in the positive *x*, *y*, or *z* directions, is an *accessible* cube. Note that this holds true whether or not there is a finite or infinite boundary plane in the negative *z* direction.

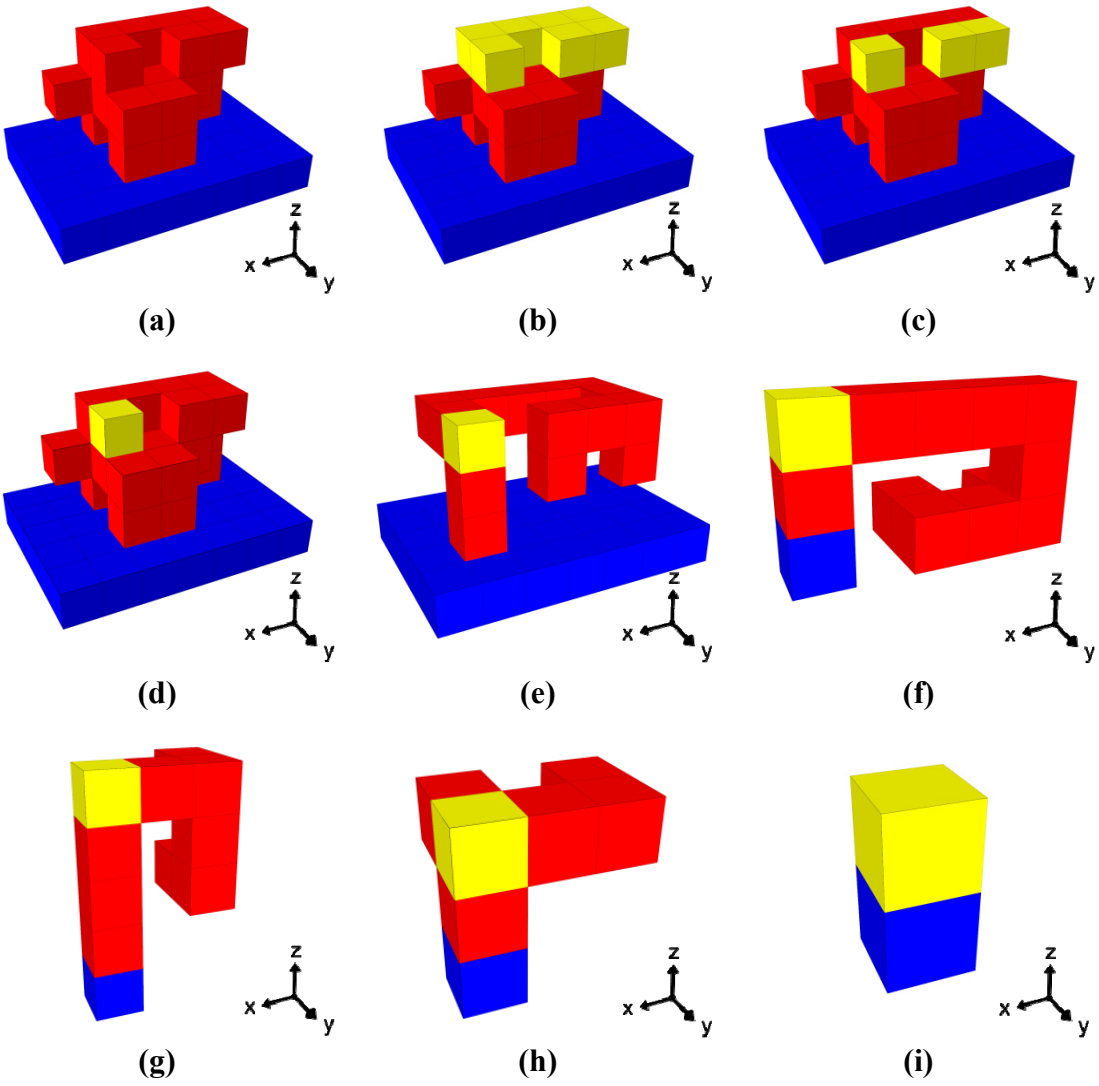


Figure 7.3. Finding an accessible cube that does not disconnect the target structure. (a) Given a target structure (red cubes) attached to a boundary plane (blue cubes), an accessible cube can be found by (b) considering the top layer of the structure, (c) moving to the front row, then (d) selecting the leftmost cube in this row. (e) If removing an accessible cube found this way (yellow) would disconnect the structure, (f) perform the same procedure on the substructure consisting of one of the substructures that would be disconnected. (g)-(i) Following this procedure recursively will, in the worst case, eventually find a single-cube substructure which, by definition will not disconnect the structure when removed.

Lemma 2. If an admissible structure can be separated into disconnected pieces by the removal of an accessible cube, these pieces, considered individually, are themselves admissible structures.

Proof. We prove Lemma 2 by contradiction. Assume one of the pieces created by removing an accessible cube from an admissible structure is not itself an admissible structure (this piece is either infinitely sized, non-contiguous, or not composed of regular cubes assembled on a rectilinear lattice). If we now replace the cube that was removed, we obtain a structure that contains a portion that is either infinitely sized, non-contiguous, or not composed of regular cubes assembled on a rectilinear lattice. Thus the original structure is inadmissible, which contradicts the assumption.

Lemma 3. Any admissible structure connected to a boundary plane will have at least one removable cube.

Proof. We prove Lemma 3 by construction. By Lemma 1, the given admissible structure must have at least one accessible cube [Figure 7.3(e)]. Removing this cube will either separate the structure into pieces disconnected from each other and the ground plane (Case 1) or not (Case 2). For Case 2, the lemma has been proven as we have found an accessible cube that can be removed without disconnecting any part of the structure from the boundary. For Case 1, we consider the substructures that would have been created if this cube were removed. At least one of these substructures is not connected directly to the boundary plane and is connected to the rest of the original structure only at the chosen accessible cube. In addition, this substructure contains at least one cube less than the original structure. By Lemma 2, this substructure is itself an admissible structure. If we consider the chosen accessible cube to be this substructure's boundary plane, we can now, by Lemma 1 find an accessible cube in

this substructure [Figure 7.3(f)]. We are now left with either Case 1 or Case 2 as described above. Each recursion of the process described above for Case 1 will reduce the size of the substructure under consideration by at least one cube [Figure 7.3(g) and (h)]. Since an admissible structure has a finite number of modules, eventually, if a Case 2 is not encountered, the substructure under consideration will consist of only one cube [Figure 7.3(i)]. Since removing a single cube that is connected to the remaining structure through a single accessible cube does not separate the structure into disconnected pieces, this cube satisfies the lemma. Thus, using the procedure we are guaranteed to find a removable cube.

Theorem 1. Algorithm 1 will find a finite, error-free assembly sequence for any given admissible structure attached to a boundary.

Proof. We prove Theorem 1 by induction. By Lemma 3, at the first stage of disassembly, there exists at least one removable cube. Thus, choosing and testing structure cubes randomly without replacement, Algorithm 1 will find a removable cube at the first stage of disassembly. With the first cube removed, the structure is now either empty (Case 1), a contiguous structure attached to the boundary (Case 2), or a set of contiguous structures each attached to the boundary (Case 3). In Case 1 we have completed a disassembly sequence. In Case 2, the remaining structure is an admissible structure attached to the boundary with one less cube than the original. In Case 3, by Lemma 2, each of the remaining structures is an admissible structure attached to the substrate, and the sum of their constituent cubes is one less than that of the original structure.

By Lemma 3, the remaining structure(s) in Case 2 and Case 3 each has at least one removable cube. The same is true for all further substructures revealed by the removal

of these cubes. Since the original structure is finite, it must contain a finite number of modules. Thus, there are a finite number of these disassembly steps required to disassemble the complete structure.

Assuming the constraints we have placed on a disassembly step are the same as those on the reverse operation (assembly), the sequence composed of the reverse of each disassembly step in reverse order is a valid assembly sequence with finite steps that contains every module in the given structure.

Sample-Based Parallel Assembly

Algorithm 1 is guaranteed to find one possible sequence for the assembly of a given target structure. However, this alone does not solve the problem of optimal stochastic assembly because waiting for a module to arrive at one particular location could take a prohibitively long time, especially if the transportation mechanism is random motion. In order to accelerate this process we would like to be able to pursue many potential assembly sequences in parallel. To achieve this, we follow a sampling approach that iterates Algorithm 1 many times to map out parallel assembly paths.

Essentially, this sampling approach repeats Algorithm 1 many times, each time starting with a different random, accessible module. During sampling, we keep track of the location of the last module removed in each sequence, as well as the sequence's expected assembly time. This sampling gives a set of possible locations to next add modules. Pursuing multiple assembly sites from among this set in parallel significantly reduces the expected assembly time because the overall expected parallel assembly rate is the sum of each of the expected serial assembly rates. For example, if a cube is expected to assemble to a single site at a rate of 1 per second, and two assembly sites are active simultaneously, then the assembly rate will double to an average of 2 cubes per second (assuming the assembly rates are independent).

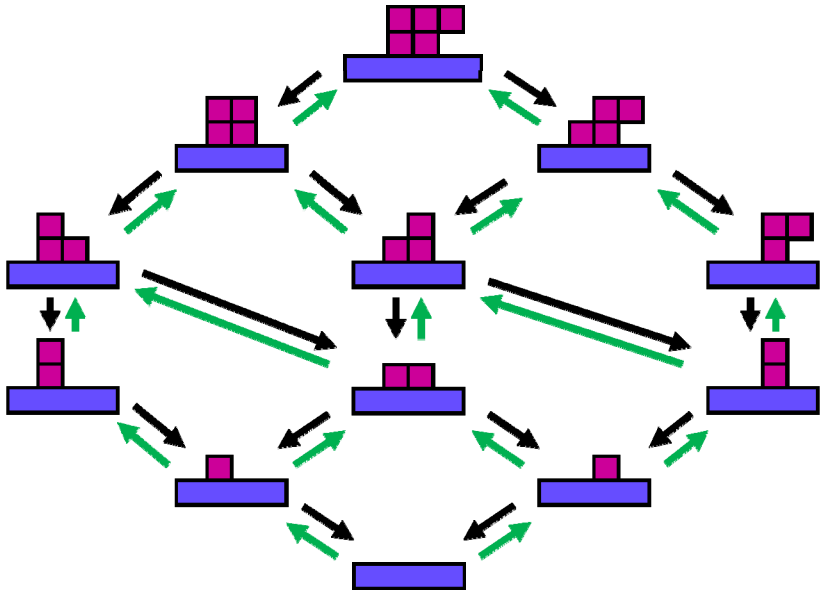


Figure 7.4. Assembly by disassembly. The parallel stochastic assembly planner samples the graph of all possible paths to assembly by starting with the complete structure and removing one cube at a time until the existing state is revealed (black arrows). Reversing the disassembly sequence reveals a valid assembly sequence (green arrows). Assembly promotion locations are chosen based on these samples.

In essence, this procedure is sampling the graph of all possible assembly sequences from the current state to the goal state (Figure 7.4). While this graph could theoretically be enumerated in its entirety, in practice it is prohibitively expensive to compute for all but the simplest structures.

The price to be paid for the increased assembly rate of this parallel assembly approach is a decrease in the certainty about the assembly sequence. In addition to not knowing the time of the next assembly event, we now also do not know its location. Thus, the stochastic process (or "nature") chooses which of the selected promotion sites will be filled at each stage of assembly. Our parallel assembly algorithm must be able to cope with this additional uncertainty. Fortunately, this can be achieved simply by performing the same sampling for each stage of assembly to choose the next set of promotion sites. Algorithm 2 details this approach.

The entire process begins with the user specifying a target shape map. Assembly then proceeds by alternating between Algorithm 2 selecting a set of assembly promotion sites, and the stochastic process (i.e. nature) selecting from among these sites, the location of the next assembly event. In stages beyond the first, Algorithm 1 considers the previously assembled modules to now be part of the fixed boundary when performing accessibility and disconnection checks. Figure 7.5 graphically depicts the entire parallel stochastic assembly process that employs this algorithm.

Algorithm 2 Parallel Stochastic Assembly Planner

At the start of assembly:
initialize *target* with user shape map

During each assembly stage:
update assembled based on current configuration
remaining ← (*target* – *assembled*) (the relative compliment)
for S (number of samples) do
  call Algorithm 1 on *remaining*
    increment *count* of final location in sample's *sequence-list*
  add sample's *cost-sum* to final location's total-*cost*
end for
select N locations to promote assembly

Stochastic Process:
Selects location of next module arrival
assembled is updated

Algorithm 2 has many desirable properties that make it suitable for stochastic assembly. First, it is an on-line approach as it re-evaluates progress at each stage of assembly. This allows the algorithm to pursue many potential assembly paths in parallel which greatly increases assembly rates. In addition, this has the potential to allow the approach to compensate for the unforeseen circumstances that may occur during stochastic assembly such as changes in the expected assembly rates or the

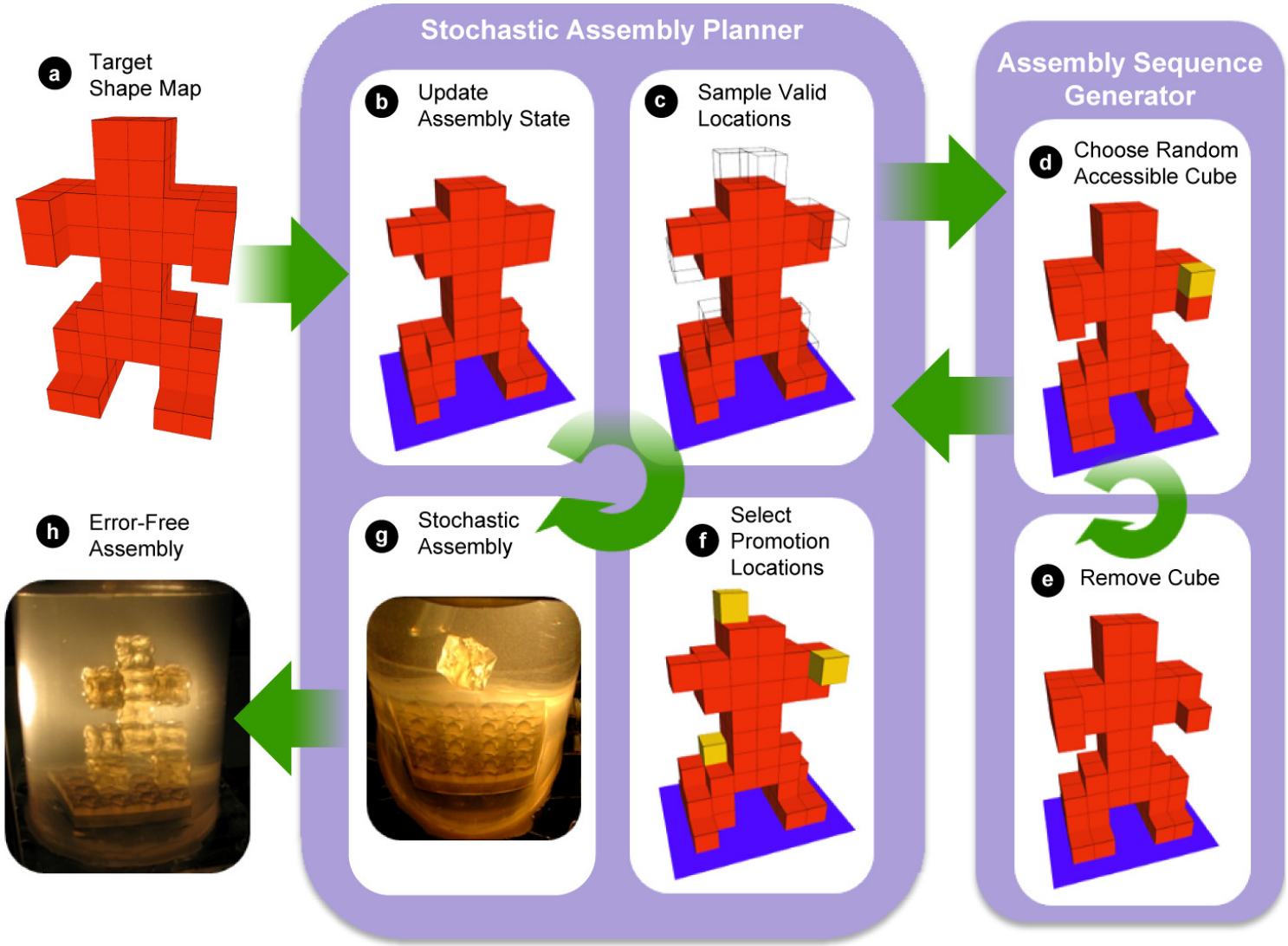
target shape. As long as the inputs to Algorithm 2 are updated it will find assembly paths based on the most recent information.

A second desirable property of Algorithm 2 is that in addition to its ability to find parallel assembly sequences, it gives us the opportunity to optimize assembly rates based on the selection of locations to allow assembly. It also gives us the opportunity to collect statistics during the sampling of the remaining structure in order to estimate the distribution of assembly rates and use this distribution to guide this optimization. We discuss potential promotion site selection heuristics in the following section.

Another advantage of the stochastic assembly planning algorithm is its potential for adaptation to different types of stochastic assembly systems, such as those shown in Figure 7.1. Below, we discuss the application of this algorithm to a Stochastic Fluidic Assembly (SFA) system. However, by tuning various parameters, such as the expected assembly rates and associated local structure configurations, as well as the number N of simultaneous attractions, we believe this algorithm could be applicable to a wide range of stochastic assembly situations. In the Simulations and Results section, we investigate the effects of changes in some of these key parameters.

Finally, this stochastic planning algorithm is able to adjust the amount of computational effort exerted during planning to adapt to the available time and resources. We expect increased sampling of the potential assembly sequences to give us a more accurate estimate of the relative merits of potential assembly paths, at the cost of increased computational effort. Our sampling approach provides a straightforward method of balancing between these tradeoffs. Such simulation tradeoffs are also discussed in the Simulations and Results section.

Figure 7.5. Diagram of parallel stochastic assembly process. (a) User inputs the target shape map. (b) The stochastic assembly planner updates the current state of assembly. (c) The assembly planner calls the assembly sequence generation algorithm to sample assembly sequences for the remaining structure. (d) The assembly sequence generator determines a valid assembly sequence by alternately choosing and (e) removing random, accessible cubes that do not disconnect the structure when removed. (f) Once the sequence generator returns a set of valid assembly locations, planner chooses from among these a set of promotion sites using some selection heuristic. (g) The stochastic process then determines which of the promotion sites next receives a module. (h) If there remain modules to be assembled, the process returns to step (b), otherwise, assembly is complete.


Stochastic Assembly Planner
Assembly Sequence Generator
a Target Shape Map
b Update Assembly State
c Sample Valid Locations
d Choose Random Accessible Cube
e Remove Cube
f Select Promotion Locations
g Stochastic Assembly
h Error-Free Assembly

Assembly Promotion Site Selection

We investigated two possible assembly promotion site selection heuristics. We used these heuristics to select the attraction locations from among the set of possible locations found during sampling, based on data collected during sampling. The *quickest paths heuristic* selects the potential modules with the quickest expected assembly times. The expected assembly times for each potential location are found by averaging the expected assembly times of all of the samples that removed the module in question last.

The second promotion site selection heuristic is the *most paths heuristic*. This heuristic is based on the conjecture that our random sampling process will tend to favor locations that maximize the number of paths to assembly. In other words, promotion sites that restrict the number of ways the remaining structure can be assembled will tend to be sampled less than those that lead to many possible assembly paths. We examine the relative effectiveness of these heuristics computationally in the Simulations and Results section.

Example Application: Stochastic Fluidic Assembly

In order to give a concrete description of how our stochastic assembly algorithm could be applied to achieve on-line assembly planning, we describe its application to the stochastic fluidic assembly system presented in Chapter 5. In this section we briefly review the relevant details of this system, please see Chapter 5 for a detailed description.

The example stochastic fluidic assembly system uses stochastic fluid motion to enable the assembly of minimalistic robotic assembly modules. A target structure is "grown" by adding components first to a planar substrate, then to the exposed surfaces

of previously assembled components. The components are homogenous, and their availability for assembly at any particular location at any point in time is determined by the stochastic motion of the fluid in the assembly chamber. An assembly promotion site can be activated by redirecting fluid flow through the internal structure to open a sink at any surface of the structure, which pulls in nearby components until one comes close enough to attach. However, there is a limit to the number of sinks that can be opened simultaneously (due to the increased flow rates required to maintain additional sinks). In addition, the attraction probability is affected by the local fluid flow, which in turn is affected by the local assembled structure. In fact, the fluid flow is highly coupled with the existing assembled cube configuration, but we assume here for simplicity that the locations immediately adjacent to a sink have the largest impact. Thus, the expected cube arrival time is a function of local geometry only.

Due to kinematic constraints (the inability to insert a module directly between two opposing modules), there are three possible configurations to which a cube can be attracted [Figure 7.6(b)]. In order to determine the probability distributions for cubes being attracted to these three basic configurations over time, we conducted a set of experiments with a test fluidic system (Figure 7.6). Each trial was conducted starting with four 1 cm cubes moving stochastically in water in a 15 cm tall by 6 cm radius cylindrical tank. The fluid agitation was achieved by pumping water out through two inlets at the top of the tank, and back in through a jet near the middle of the tank at approximately 1.5 L/min [as indicated in Figure 7.6(a)]. At a random point in time, a valve was opened to allow fluid to flow out through the sink(s) at the target location. We then measured the time until one of the free-floating cubes to be attracted to the target location. This experiment was repeated 50 times for each of the three configurations.

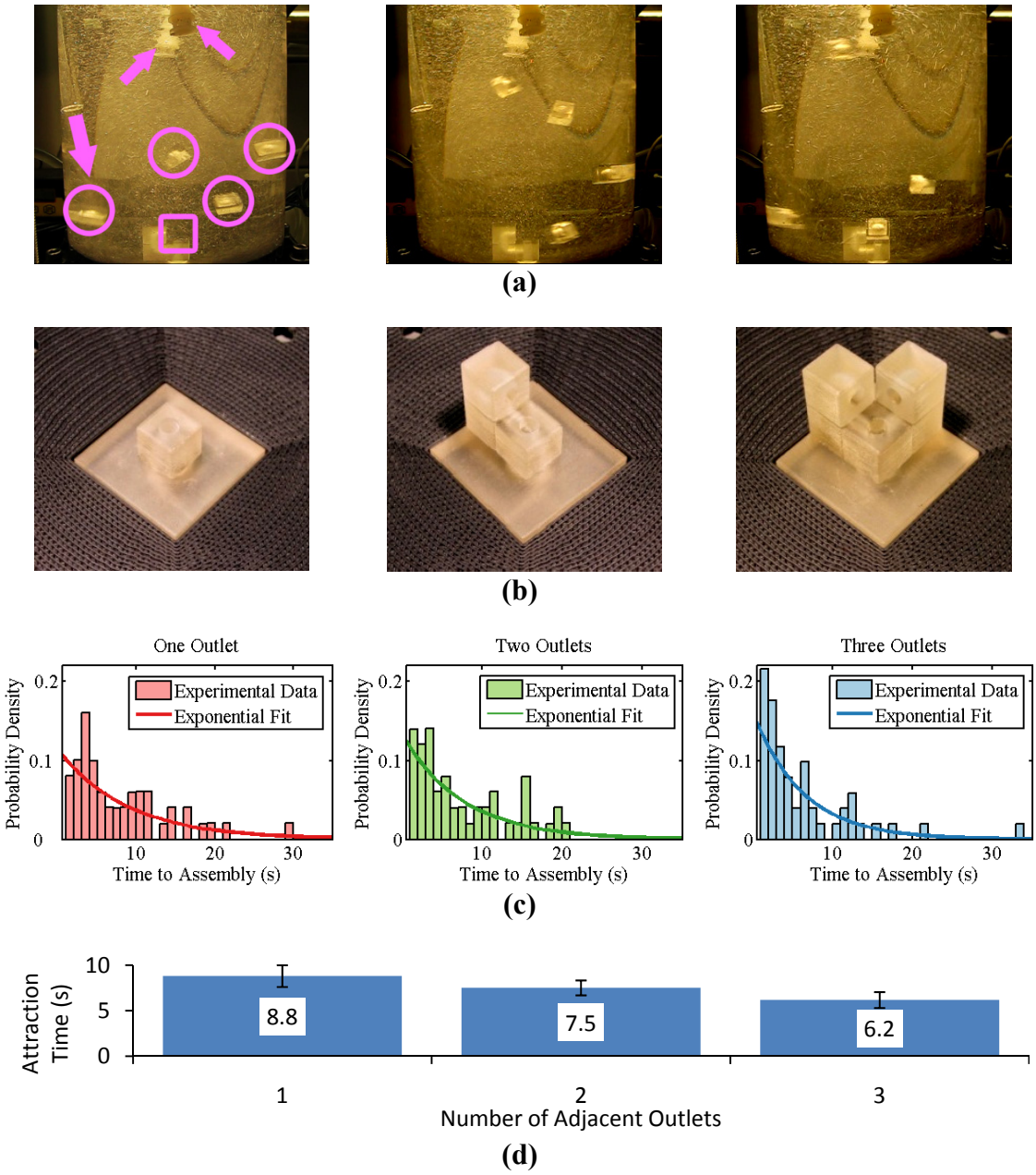


Figure 7.6. Experimental determination of assembly probability function. (a) Experiments were conducted to record the time required to attract a free module undergoing stochastic agitation to one of three outlet configurations. Large arrow indicates inflow, smaller arrows indicate outflow, circles identify 1 cm cubes undergoing stochastic agitation, square denotes target location to which cubes are attracted. (b) The three possible outlet configurations to which a new module can be attached. (c) Plot of mean time to attraction over 50 experiments for the three configurations. (d) Comparison of the mean assembly times for the three cases, with the standard errors indicated.

The results of these experiments are shown in Figure 7.6(c), which shows histograms of the durations of the trials, binned into one second intervals up to 35 s. We see that the assembly times were roughly exponentially distributed, with fast assembly times being more likely, but much longer assembly times also being possible. Overlaid on the experimental results are the most probably exponential fits. Figure 7.6(d) compares the mean assembly times for the three cases.

It is important to note that these experiments investigate only the time required to attract a module to the right location and do not take into account any additional time required to engage latching mechanisms and establish communication between modules. Nonetheless, these results support our assumption that the assembly rates in our system depend on the local structure configuration. Additionally, they provide us with a lower bound on the assembly time distributions for testing our assembly algorithm.

Using these three expected assembly times, we can create a simplified model of our system for collecting statistics while varying the parameters and selection heuristics of our path planning algorithm. The fundamental question is how to sample, at any stage of assembly, which of a set of the exponentially distributed assembly events occurs next and when it occurs. Drawing inspiration from chemistry, we can use a method developed by Gillespie (1977) to sample from a set of potential simultaneous reactions in a well-mixed stochastic fluidic environment with multiple chemical species (Gibson & Bruck, 2000). Gillespie's method relies on the following probability density that the next of j possible reactions is μ and that it occurs at time τ:

$$P(\mu,\tau)d\tau = a_\mu exp\left(-\tau\sum_j a_j\right)d\tau \qquad (7.1)$$

where a_j is the rate coefficient of reaction j (that can depend on chemical concentrations, environmental conditions, etc.). Integrating $P(\mu,\tau)$ from $\tau = 0$ to ∞

gives the probability that the next reaction to occur is a given reaction μ:

$$P(\mu) = \frac{a_\mu}{\sum_j a_j} \qquad (7.2)$$

Further, summing $P(\mu, \tau)$ over all reactions μ gives the distribution of times for the next reaction to occur:

$$P(\tau)d\tau = \left(\sum_j a_j\right) exp\left(-\tau \sum_j a_j\right) d\tau \quad (7.3)$$

Integrating this equation and solving for the time τ as a function of a selected random number P in the range (0,1) gives an equation for randomly sampling the time taken for the next equation to occur:

$$\tau = -\frac{\ln(1-P)}{\sum_j a_j} \qquad (7.4)$$

Thus, if we view the various parallel potential assembly events that could occur at any point in time as chemical reactions in a well-mixed system, we can apply equations (2) and (4) to determine the time and location of the next cube arrival in our stochastic assembly system. In this case, the rate constants can be determined by inverting the mean times to assembly from our experimental results shown in Figure 7.6.

Simulations and Results

We employed the method described in the previous section to simulate our stochastic fluid assembly system. This simulation proceeds by alternating between two steps, as described in Algorithm 2. After initialization of the assembly planner with the desired user shape map, in the first step of the simulation, the planner finds a set of allowable promotion sites and selects a subset of these to promote assembly. The second step of the simulation uses Gillespie's method described in the previous section to randomly select the time and location of the next assembly event, according

to the distributions determined by the expected assembly times of the locations in question. These expected assembly times, in turn, are determined by the local structure configuration. The simulation proceeds by alternating between these two steps of assembly planning and random selection until the structure is complete. In the rest of this section we describe results of our investigations into the effects of changing the location selection heuristic, as well as the number of parallel promotion sites and path samples for each assembly stage. Finally, we analyze the scaling of the computational effort exerted by our algorithm with the number of modules in the target structure.

Effect of promotion site selection criteria

We ran the simulation described above 50 times for a human hand target shape composed of 619 modules [Figure 7.7(a)]. We set the number of simultaneous attractions to four and sampled the remaining structure 40 times at each assembly stage. We used the experimentally determined expected assembly times for the three assembly cases and calculated the total assembly time for each run. Figure 7.7(b) shows a comparison of the mean assembly time per cube (averaged over 50 runs), for the two promotion site selection heuristics described in Section 4, along with a random baseline (where assembly promotion locations are chosen at random from among the valid locations found by the assembly planner). Much to our surprise, we found no statistical improvement (decrease) in the assembly times for the heuristics as compared to the random baseline.

One possibility for this null result in the effectiveness of the promotion site selection heuristics was that their improvement was simply lost in the noise of the stochastic simulation. The similar experimental assembly time values for the three location configurations [Figure 7.6(b)] could have made the reductions in the assembly times undetectable. In order to test this, we prescribed a new set of expected

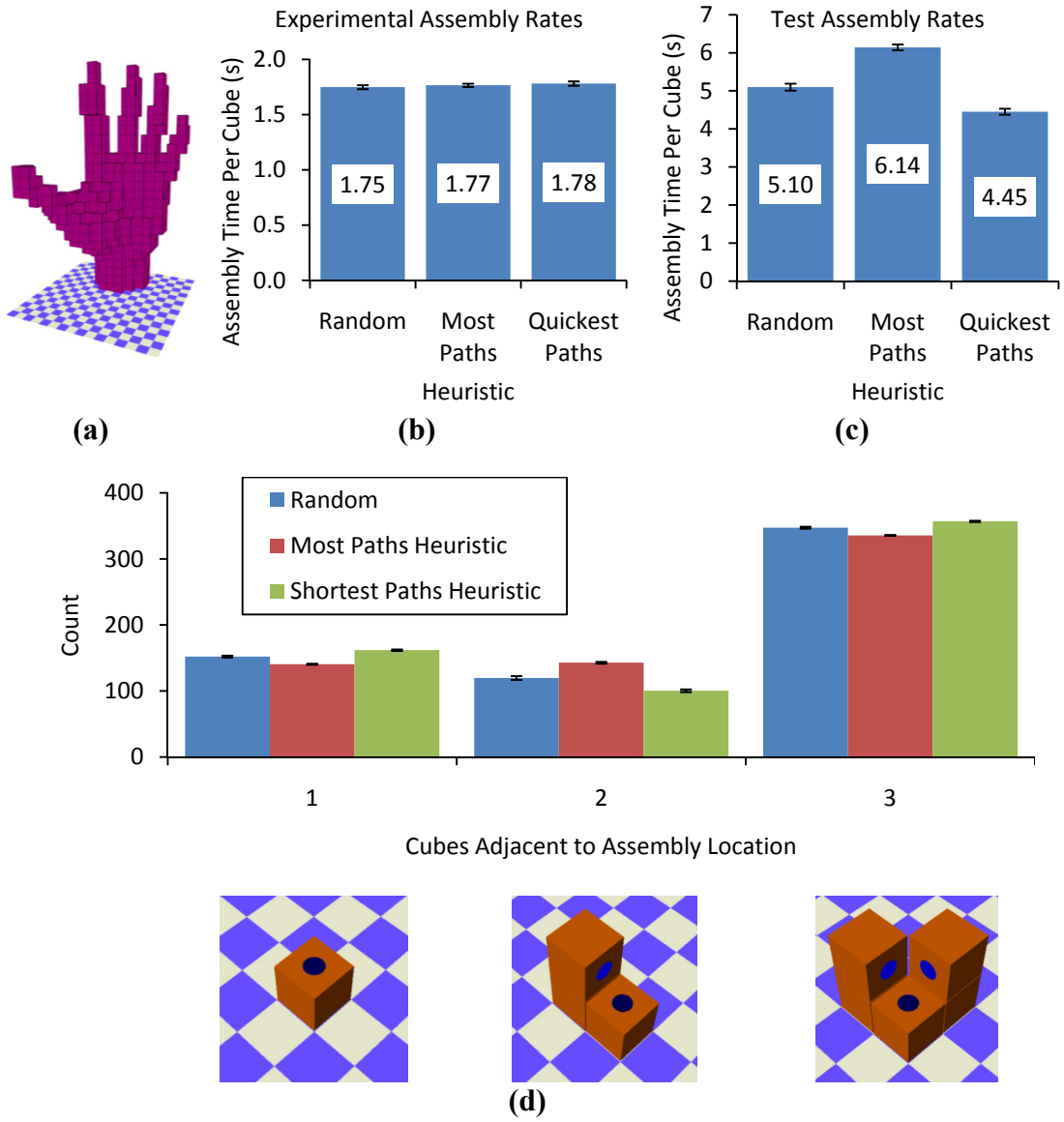


Figure 7.7. Assembly promotion site selection. Two heuristics for choosing the subset of viable sites at which to encourage assembly were tested on the assembly of a 619-module hand-shaped target (a). (b) Simulations using experimental assembly rates found no improvement with these heuristics as compared to random site selection. (c) A larger discrepancy in the assembly rates, however, reveals a significant time reduction for the Quickest Paths heuristic, and an increase for the Most Paths heuristic. (d) A breakdown of the numbers of each of the three possible neighboring module configurations reveals the cause of the results seen in (c).

assembly times: 1s for Case 1, 100 s for Case 2, and 1 s for Case 3. Essentially this simulates a situation where, for some reason, it has become very expensive (in terms of time) to attach a module to a location with exactly two neighbors. This could be due to some practical constraint in the system due to the latching mechanism, for example.

The assembly time comparison for this new situation can be seen in Figure 7.7(c). We now see a statistical difference caused by the selection heuristics, however, again there is a surprise. The *Quickest Paths* heuristic reduced assembly times as compared to random, but the *Most Paths* heuristic actually increased assembly times.

A closer look at the ratios of the three assembly configurations encountered by the different heuristics helps to understand their relative performance. Figure 7.7(d) shows the average number of modules assembled to each of the three possible configurations over the course of 100 simulations for the three different heuristics. Using the *Quickest Paths* heuristic has the effect of reducing the number of assemblies to the expensive configuration (two adjacent), as compared to selecting the promotion sites randomly. By contrast, the *Most Paths* heuristic – which does not take assembly times into account – actually increases the frequency of the expensive case. Thus, any benefits due to maximizing the number of assembly steps with the maximum number of promotion sites are outweighed, at least in this case, by the expensive assembly rate of the assembly configurations preferred by this maximization.

Effect of target shape

In order to test if the relative effectiveness of the promotion site selection heuristics was not specific to the chosen target shape, we conducted the same experiment in simulation on four other target shapes of various sizes (Figure 7.8). In each case, the average assembly time when following the *Quickest Paths* heuristic was significantly lower than when following either the *Most Paths* or *Random* alternatives.

Also, in each case, the *Most Paths* heuristic's average assembly time was higher than random, except for the *Eiffel* shape, which was not significantly different.

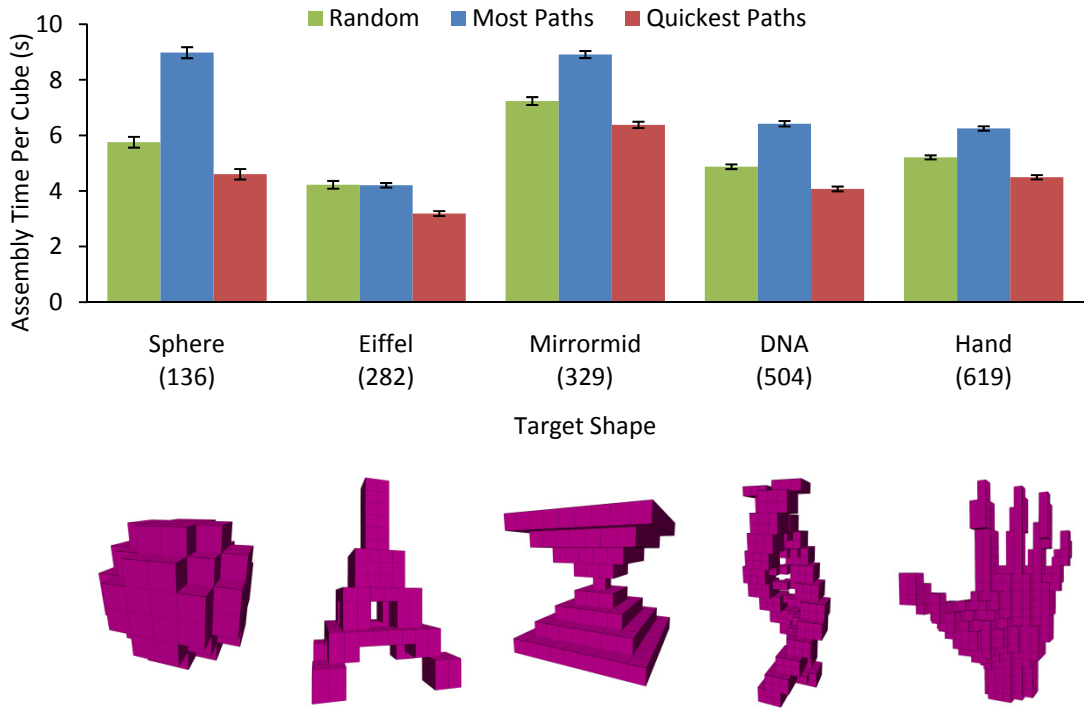


Figure 7.8. Effect of target shape. The simulations performed to determine the effectiveness of the various promotion site selection heuristics on the assembly of the hand target shape (Figure 7.7) were repeated with four other targets of varying shapes and sizes. The numbers in brackets indicate the number of modules in each shape. The relative effectiveness of each of the three promotion location selection heuristics was similar in each case.

Effect of number of samples

Since our assembly planner re-samples the remaining structure at each stage of assembly, the number of samples of the remaining structure performed at each stage of assembly obviously has a large impact on the computational effort exerted during assembly planning. Thus, we would like to know how many samples are necessary to realize an (at least locally) optimal assembly time. In order to investigate this, we ran simulations as described above, with four simultaneous promotion sites and the

tweaked assembly costs, and varied the number of samples. In Figure 7.9(a) we compare the mean assembly time per cube, averaged over 100 runs, for cases with 10, 20, 40, and 60 samples per assembly stage, for the *Quickest Path* heuristic versus random promotion site selection.

The first thing we see from this comparison is that the *Quickest Path* heuristic outperformed random location selection in each case. This improvement does not seem to be dependent on the number of samples. However, we also see that the average assembly time per cube decreases as we increase the number of samples from 10 to 40 samples, *even with random location selection*. This can be explained by the fact that 40 samples are more likely than 10 to find at least four different potential promotion sites at each assembly stage. Any time the number of distinct potential locations found is less than the allowed number of parallel promotion sites (in this case four), the overall assembly rate will suffer. However, this effect seems to have diminished as the number of samples increased from 40 to 60, indicating that there is an ideal number of samples (in this case approximately 40), above which the additional computational effort does not result in reduced assembly times.

Effect of number parallel promotion sites

We were also interested in the effect of the number of parallel assembly promotion sites. While this parameter is most likely determined by practical system limitations, it may be useful to know how beneficial it would be to increase the number of parallel promotion sites. To this end, we conducted another set of simulations in which we varied the number of parallel promotion sites. For these simulations, we kept a constant ratio of 10 samples per promotion site. Figure 7.9(b) is a plot of the results.

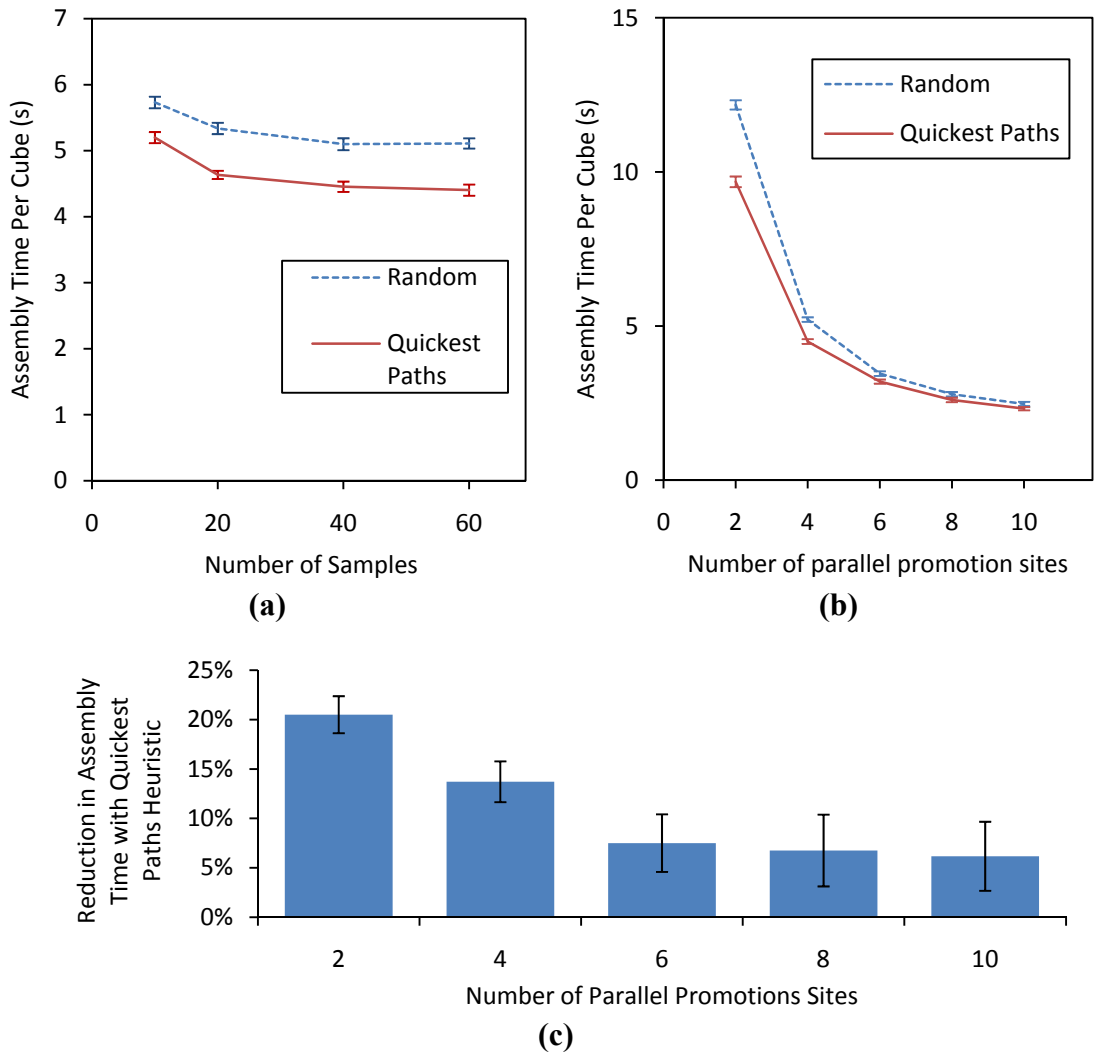


Figure 7.9. Effect of number of samples and parallel promotion sites. (a) Increasing the number of samples decreased the assembly time per cube, up to a certain limit. (b) Increasing the number of parallel promotion sites decreased the assembly time per cube, also with diminishing returns. (c) Beyond four parallel promotion sites, the reduction in assembly time due to the Quickest Paths heuristic remained relatively constant at approximately 7%.

As expected, the average assembly time per cube decreases as the allowed number of parallel sites increases. This is, of course, the reason for pursuing parallel assembly to begin with. However, we see diminishing returns as the number of promotion sites increases. This is most likely due to the finite number of potential promotion sites at any given stage of assembly, as determined by the target shape. (I.e. increasing from

six to eight allowed promotion sites doesn't help if you're assembling a structure with a maximum of six valid promotion sites at any given assembly stage.)

Another interesting feature of the data shown in Figure 7.9(b) is the decrease in the assembly time reduction due to the *Quickest Paths* heuristic. Figure 7.9(c) shows this reduction as a percentage of the overall assembly time. We see that this reduction decreases from 20% for the two promotion site case down to 5% for the 10 site case. Thus it seems that the advantage of choosing the promotion sites carefully decreases as fraction of the total possible sites chosen increases. However, the time reduction at 10 sites is still significant in this case.

Estimating Assembly Time

In many situations we would like our on-line assembly algorithm to be able to predict the amount of assembly time remaining and to give some upper and lower confidence bounds on this estimate. For a serial assembly path, the estimate can be found simply by summing the expected assembly times for each cube addition along the path. However, we would like to be able to estimate the average time required to assemble a target structure while attracting at multiple locations in parallel, based on the sampling of possible serial assembly paths.

Expected Assembly Time

We estimate the expected assembly rate for the case of multiple modules being attracted in parallel by summing the sampled serial assembly rates. For example, if we are attracting modules to three locations in parallel, and our sampling indicates that the average expected serial assembly times for the rest of the structure starting with these three locations are 8, 9, and 10 s respectively, their combined parallel assembly rate would be:

$$Parallel\ Assembly\ Rate = \frac{1}{8\ s} + \frac{1}{9\ s} + \frac{1}{10\ s} = 0.336\ s^{-1} \qquad (7.5)$$

Thus, the expected assembly time would be:

$$Expected\ Assembly\ Time = \frac{1}{Parallel\ Assembly\ Rate} = 2.98\ s \qquad (7.6)$$

Figure 7.10 shows a plot of the estimated and simulated remaining assembly time for each stage of the parallel assembly of a 104-module sphere target shape. While there can be a large amount of variation in the actual (or simulated) assembly time, we see that, at least in this case, the estimates based on the serial samples of the remaining structure give a reasonable prediction of the assembly time remaining.

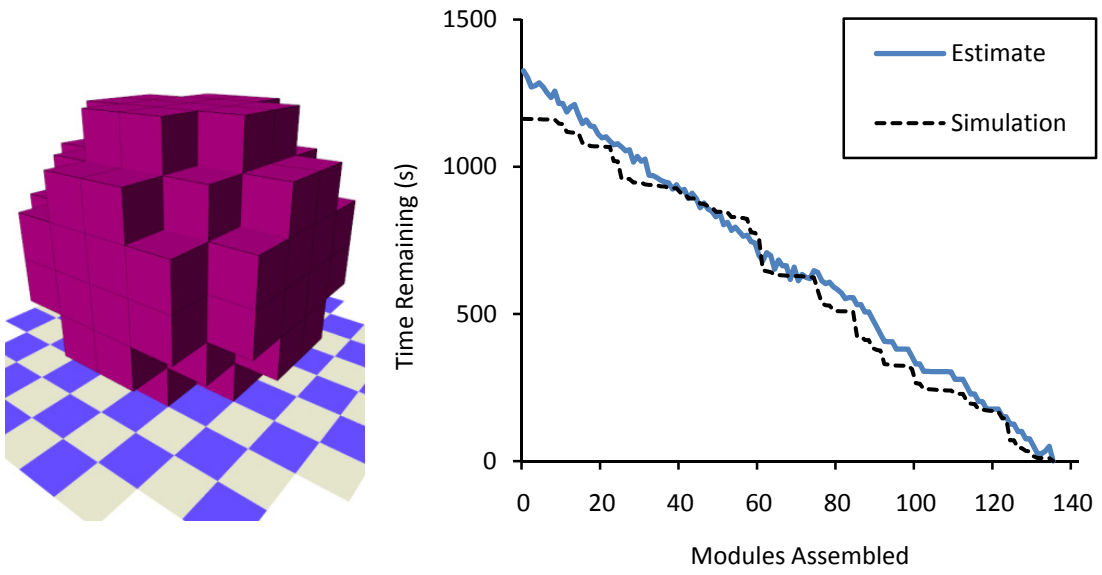


Figure 7.10. Estimating the expected assembly time for a 136-module sphere target shape. The expected assembly time for the remaining structure is estimated based on serial samples of the remaining structure to be assembled which are combined into an estimate of the parallel assembly time.

Determining a confidence interval

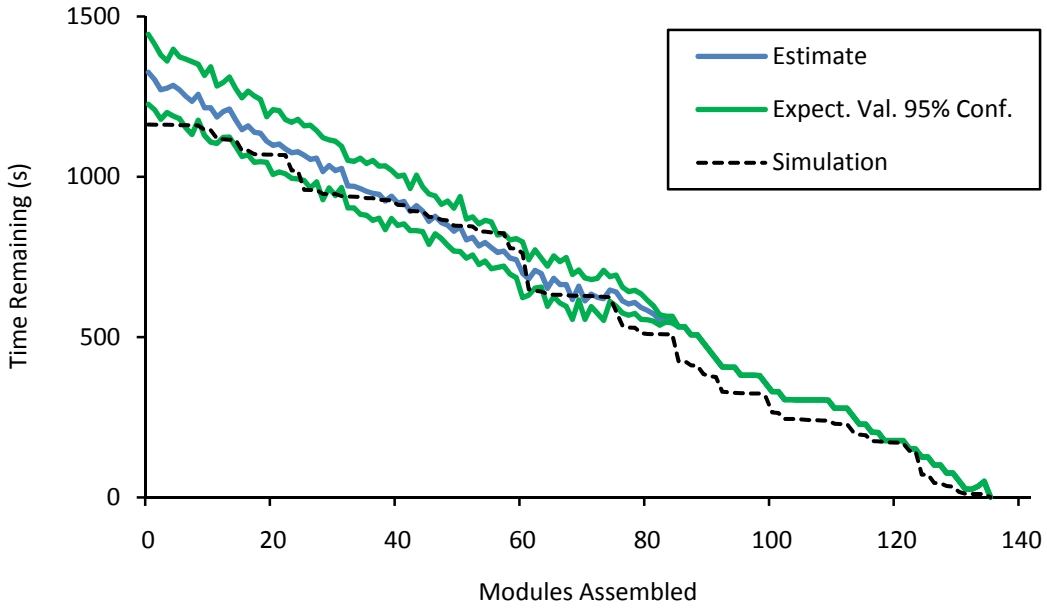
As we see in Figure 7.10, for any particular run there is some discrepancy between expected and simulated assembly time. We would like to be able to determine upper

and lower limits that we can be reasonable confident (i.e. 95%) that the time of any individual assembly experiment will fall between. Two of the factors that contribute to the error in our estimate are the uncertainty in the expected assembly time and the variation due to the random, exponentially distributed arrival times of the modules. We deal with these two uncertainties separately in the next two sections.

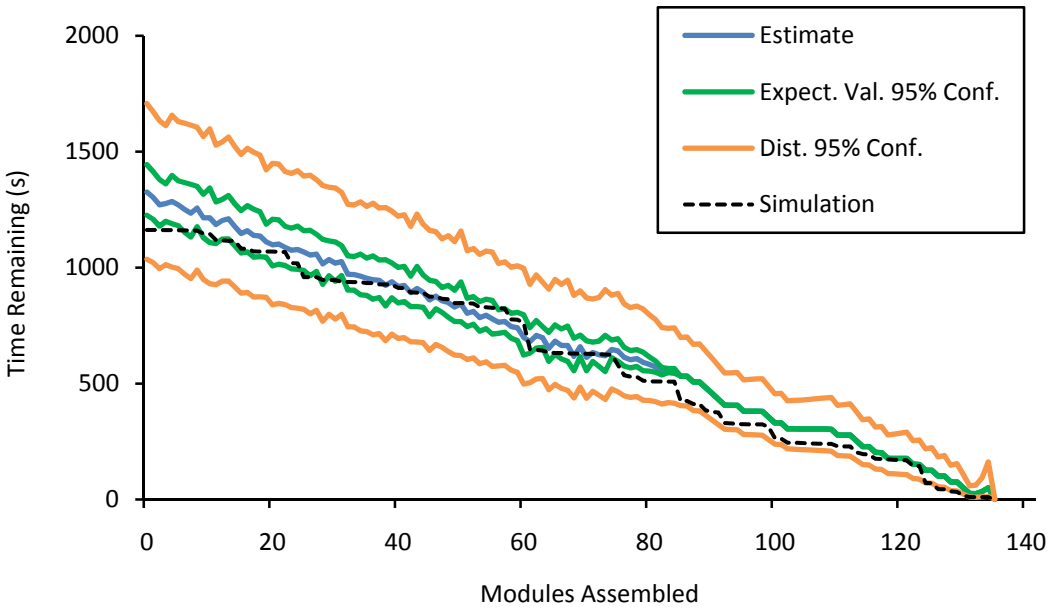
Expected Time Uncertainty

The uncertainty in the expected assembly time is due to the fact that this value is calculated from a set of samples of the structure, conducted as described above in the description of Algorithm 2. The parallel assembly rate is obtained by summing a number of serial assembly rates, each of which is estimated based on the number of random samples that end at the chosen locations. If we assume that these rates are each normally distributed and independent, we can calculate and sum their variations in order to obtain the standard deviation of the expected time. Figure 7.11(a) shows the upper and lower 95% confidence interval (1.96 standard deviations) for the expected value.

One challenge in this calculation is that our algorithm calculates a predetermined number of samples in order to choose the optimal set of locations to attract modules (40 samples were used in this case). Additionally, there is no way of specifying which locations are associated with each sample (since the paths are chosen randomly). Thus, any particular location may end up with a small, or large, number of samples. This makes it difficult to determine the variance of the associated assembly rate. This problem could potentially be alleviated by continuing sampling until each chosen location has a minimum number of samples, or while time permits.



(a)



(b)

Figure 7.11. Calculating confidence bounds on the range of possible assembly times for our assembly planning algorithm. (a) One component of the overall uncertainty is due to uncertainty in the expected assembly time calculated based on a limited number of samples of the remaining structure to be assembled. (b) A second contribution is due to the random, exponentially distributed arrival times of the individual modules.

Arrival Time Distribution

While we would expect a large number of repeated assembly experiments to approach the expected value estimated as above, it would in fact be quite surprising if this were the case for a single experiment. This is because each module is arriving at a random, exponentially distributed time. The distribution of the sum of a series of independent, exponentially distributed times is itself a gamma distribution. From the gamma distribution we can obtain 95% confidence bounds for the structure assembly time based on the expected values estimated as described in the previous section.

In Figure 7.11(b) we see the 95% confidence bounds from the gamma distribution for the upper and lower bounds of the expected value from Figure 7.11(a). For this particular run, we see that the simulated value falls within these bounds for the majority of the assembly. We can see from Figure 7.11(b) that a large part of the uncertainty in the overall structure assembly time comes from the random distribution of module arrival times. Nonetheless, the uncertainty in the expected value is also significant.

Evaluation of Assembly Time Estimates

We tested our assembly time predictions over the course of 120 runs of the 619-module hand target shape [as seen in Figure 7.7(a)] for both the *experiment* and *test* single-module assembly time values described in the *Simulations and Results* section. (*Experimentally*, we found the expected assembly times for cube arrivals to be 6.2 s, 7.5 s, and 8.8 s while for the *test* case, we prescribed them to be 1 s, 100 s, and 1 s.) The results of these simulations can be seen in Figure 7.12. Part (a) shows the estimated assembly time remaining at each stage of assembly, as well as 95% confidence bounds and actual simulated assembly time for one particular run with the *experimental* values. Note that the 95% confidence bounds on the estimate overlap

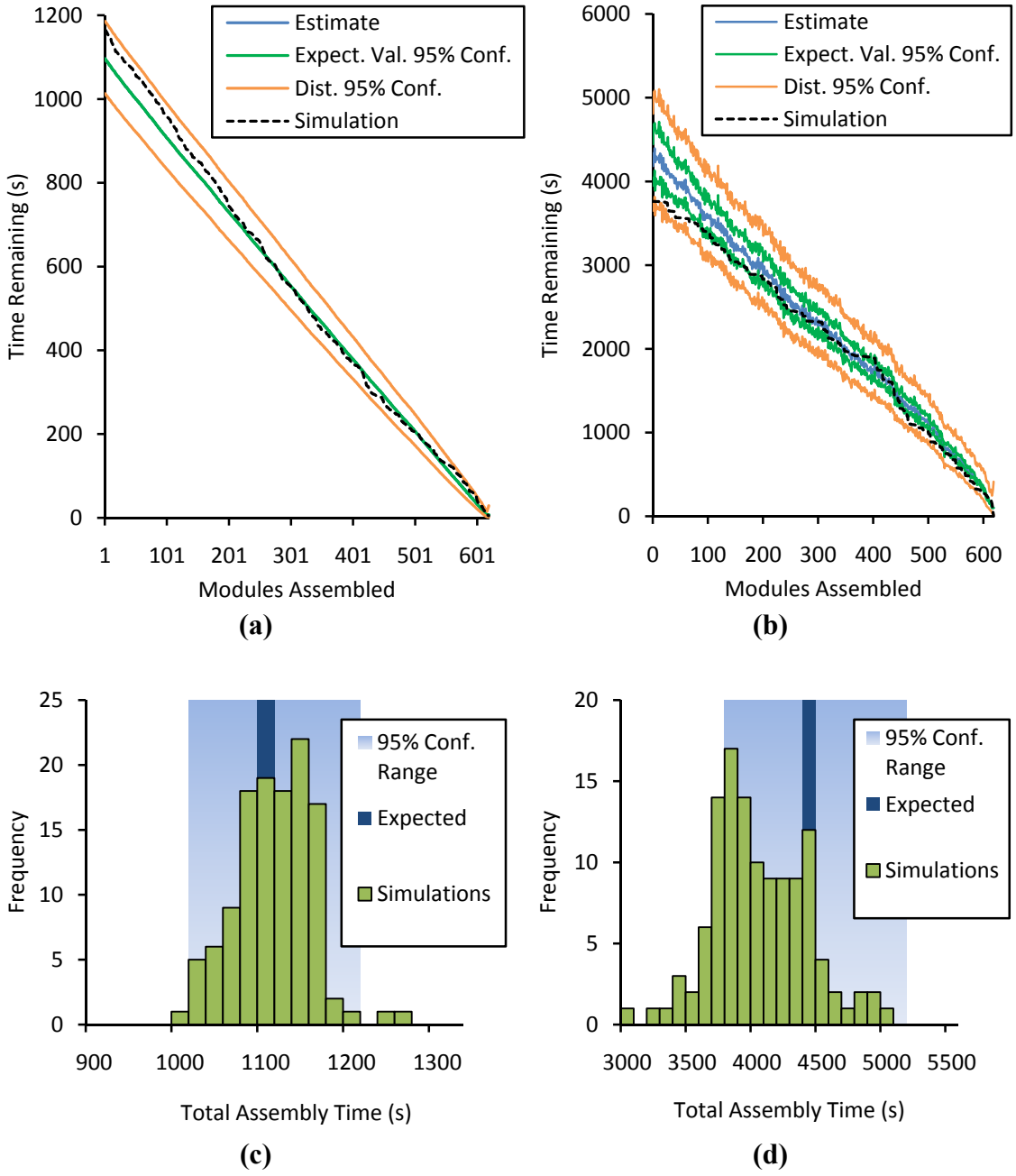


Figure 7.12. Assembly time prediction confidence bounds. (a) plot of estimated and simulated time to assembly for the 619-module hand shape [see Figure 7.7(a)] using the *experimental* cube assembly times described in the *Simulations and Results* section above. (b) same as (a) except using the *test* assembly times. (c) Histogram of simulated total assembly times over 120 runs for the hand shape using the *experimental* cube assembly times. Also indicated are the average expected assembly time and the average 95% confidence range. (d) same as (c) except using the *test* assembly times.

the estimate itself due to the similarity of the three experimental expected cube arrival times. Part (b) shows the same plot for the *test* values for which one case was much more expensive than the others. Here we see a much larger range for the confidence bounds (even after accounting for the much longer overall assembly times).

Figure 7.12(c) and (d) shows histograms of the results of similar simulations repeated 120 times for both assembly rate cases. The bars represent the number of simulations (out of 120) with total assembly times within the indicated ranges [20 s ranges in (c) and 100 s ranges in (d)]. The bin containing the average of the expected times calculated in the 120 simulations is also indicated, as well as the average range of the 95% confidence bounds. In the first case [Figure 7.12(c)], 97.5% of the simulations assembly times fell within the 95% confidence range. In addition, the average simulated time (1100 s) was very close to the average estimate (1096 s). However, in the second case [Figure 7.12(d)], the average simulated time (3964 s) was a fair bit lower than the average estimate (4312 s) and only 76.7% of the simulation fell with the confidence range. Interestingly, if the expected value were corrected (i.e. if 348 s was subtracted from all predictions), then 93.3% of the simulations would fall within the 95% confidence range. This suggests that although the algorithm systematically over-predicted assembly times for the test case, it did give a reasonable confidence interval for the predictions.

Computational Effort and Scaling

We studied the scaling of the simulated assembly time and the total assembly algorithm computation time as a function of the number of modules in the target structure (Figure 7.13). Each data point represents 20 simulated assemblies of target structures of various sizes and topologies (including those seen in Figure 7.8). The *CPU Time* was the total time taken to run the entire assembly simulation on one core

of an Intel Core 2 Quad Q6600 processor.

As expected, the simulated structure assembly time scales linearly with the number of modules, and the data fits a linear curve with a coefficient of determination of 0.991. As for the assembly planner, we can first predict some analytical bounds on its scaling. In the worst case, we expect the computation requirements of the assembly planner to scale with $O(kn^4)$, where n is the number of cubes in the structure and k is the number of samples performed at each step. This is because in the worst case the sequence generator (Algorithm 1) must visit every cube in the structure during the connectivity check, and in finding an accessible cube, for each disassembly step. Finally, Algorithm 2 calls Algorithm 1 k times for each cube in the target structure.

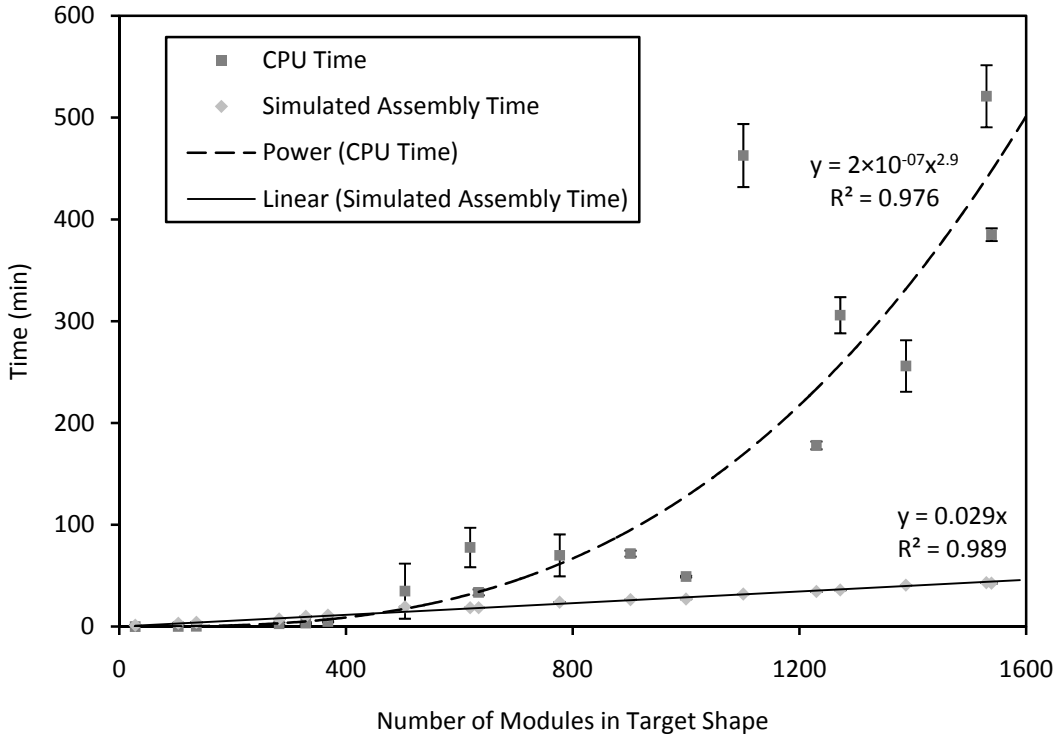


Figure 7.13. Assembly planning algorithm scaling. The time taken to assemble a structure in simulation scales linearly with the number of modules while the total time taken to compute the locations to next attract modules at each stage scales with approximately the cube of the number of modules.

For a lower bound, we assume that Algorithm 1 finds an accessible cube and checks it for connectivity in a single step each time. In this case, Algorithm 1 scales with $O(n)$ and thus the overall assembly planner scales with $O(kn^2)$. In practice, we would expect the time complexity of the assembly planner to scale somewhere between these bounds. In fact, if we fit the results of our empirical study to an exponential curve, we find that the computation time scales with $O(n^{2.9})$ (R^2=0.976).

Discussion

The previous section highlights one of the challenges of our proposed algorithm. The cost of using a sample-based online algorithm that is guaranteed to find an error-free assembly path is a high time complexity. For structures consisting of many modules, the assembly rate becomes limited by the assembly planner's computation as opposed to the physical system. Our results predict that this situation would be encountered with a target structure of greater than approximately 1000 modules.

Despite this challenge, there are many factors that could improve the scaling of our proposed algorithm or otherwise allow it to be useful for larger target structures. First, as mentioned previously, the simulated assembly time is a lower bound on the assembly time of a stochastic fluidic assembly system because it does not take into account the time required for latching and establishing communication. The fraction of assembly time used for planning will decrease as these considerations are included in the physical assembly time.

Second, aside from using faster processors or multi-threading, there are many ways the experimental code could be optimized for speed. First, the time complexity of the algorithm could potentially be reduced by choosing cubes to check for accessibility in a less naïve way (e.g. considering only surface cubes). Also, for cases where the gains due to optimal promotion site selection are small or nonexistent [e.g.,

for the case shown in Figure 7.7(b)], the number of samples required could be greatly reduced by simply choosing the first N distinct promotion locations (where N is the maximum number of promotion locations). Additionally, since our algorithm is re-sampling the same overall assembly graph over and over again, it may be possible to employ dynamic programming techniques to reduce the algorithmic complexity by re-using previous samples.

A third option for improving the scaling of our algorithm would be to employ a hierarchical assembly approach as described in Chapter 3 and Chapter 4. A hierarchical approach would first assemble individual modules into sub-components, which are subsequently assembled into progressively larger sub-assemblies until the final structure is complete. This approach would limit the size of the structure to be sampled by our algorithm. It would also have the advantage of allowing the assembly of multiple components in parallel, further accelerating assembly. However, this approach would require further, higher-level assembly planning, and the ability to manipulate larger sub-components. We leave these considerations for future work.

Conclusions

We have presented an approach that provably finds assembly sequences for any finite structure (Algorithm 1). Using this algorithm repeatedly to sample assembly sequences, we have developed an approach to parallel assembly applicable to a variety of stochastic robotic systems. The on-line nature of this approach makes it adaptable to many situations and changing inputs or environments. We have demonstrated the application of this approach to the assembly of structures with various topologies in simulation. In addition, we have investigated promotion selection heuristics for optimizing assembly times. We have also examined the trade-offs of varying primary algorithm parameters (the number of samples and the number of parallel promotion

sites). Finally, our scaling analysis suggests that this algorithm scales to the assembly of at least 1000 modules, although we believe there are many ways this scaling could be improved.

In addition to the directions proposed in the previous section for reducing the time complexity of the planning algorithm, we foresee many other potential areas for future work. One such area would be to investigate scaffolding by allowing modules to be removed as well as added during assembly. Error identification and correction would also certainly be required for a practical stochastic assembly system. It would also be interesting to investigate adding further constraints during assembly, such as distributing promotion sites over the surface of the structure when possible and accounting for the structural properties of substructures during assembly. Furthermore, taking more of the local configuration into account when calculating assembly rates would most likely improve the accuracy of assembly simulations. Finally, we await the opportunity to test this stochastic assembly approach on an experimental system.

CONTRIBUTIONS

Major Contributions

Chapter 1

- Presented the first known system capable of the dynamically programmable fluidic microassembly of arbitrarily defined structures.
- Demonstrated the automated deterministic assembly of structures composed of up to three components.
- Demonstrated the first known example of dynamically programmable fluidic assembly (at any scale) of structures composed of up to 10 components.
- Explored a range of microfluidic chip and tile designs, materials, assembly fluids, and fabrication processes to identify a set appropriate for dynamically programmable fluidic Microassembly.

Chapter 2

- Performed the only known analytical and experimental study of latch designs for in-plane latching of MEMS components.
- Developed a procedure for patterning microtiles with electrodes to establish an electrical connection between tiles when fluidically assembled.

Chapter 3

- Identified the capability of hierarchical assembly as a key component of future lab-on-a-chip systems.
- Demonstrated the first known example of dynamically programmable hierarchical microassembly of structures composed of up to eight components.
- Presented a new system and module design for the hierarchical fluidic

microassembly of structures composed of more than eight components.

Chapter 4

- Designed and implemented the smallest experimental 3-D stochastic fluidic assembly system known to date.
- Demonstrated module manipulation, assembly, structure repair, and hierarchical assembly in experiment.
- Evaluated robustness of 3-D assembly by collecting statistics for 10 consecutive experiments. 60% were successful and took an average of 442 s.

Chapter 5

- Developed a computationally efficient simulator to predict assembly statistics for stochastic fluidic assembly. Verified basic simulation predictions against experimental results and CFD simulations.
- Evaluated the importance of various stochastic assembly system parameters based on simulation results.

Chapter 6

- Developed a set of static control strategies for stochastic fluidic assembly and tested their various merits in simulation.

Chapter 7

- Presented a novel on-line assembly planning algorithm for stochastic assembly and proved that it will find an assembly sequence for any contiguous shape.
- Examined the effects of various algorithm parameters on assembling structures of various shapes in simulation.
- Determined a method for finding confidence intervals for assembly time predictions.

Contributions of Others

Chapter 1

- Collaborated with Ph. D. student Mekala Krishnan, who developed simulations of the microfluidic assembly system and created an analytical model, as described in the *Analytical Model and Simulations* section

Chapter 2

- Collaborated with CNF REU student Andrew Baisch on the design, fabrication, and testing of microtiles with electrodes.

Chapter 4

- Module design was inspired by the design of Ph. D. student Victor Zykov's fluidic assembly cubes (Zykov & Lipson, 2007).

Chapter 5

- Collaborated with Michael Kalontarov on the FLOW-3D CFD simulation results used to compare against the custom simulation results (*CFD and Experimental Validation of Module Attraction* section).
- Collaborated with M. Eng. student Ryan C. Lovrien on development of a stable fluids-based simulator for *Fluidic Tetris* game.

REFERENCES

Ayanian, N., White, P. J., Halaz, A., Yim, M., & Kumar, V. (2008). Stochastic control for self-assembly of XBots. *Proceedings of ASME 2008 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, (pp. 1169-1176). New York, NY.

Böhringer, K. F., Fearing, R. S., & Goldberg, K. Y. (1999). Microassembly. In S. Y. Nof (Ed.), *Handbook of Industrial Robotics* (pp. 1045-1066). New York: Wiley.

Bowden, N., Terfort, A., Carbeck, J., & Whitesides, G. M. (1997). Self-assembly of mesoscale objects into ordered two-dimensional arrays. *Science*, *276*, 233-235.

Castano, A., Behar, A., & Will, P. (2002). The Conro modules for reconfigurable robots. *IEEE/ASME Transations on Mechatronics*, *7* (4), 403–409.

Chung, S. E., Park, W., Shin, S., Lee, S. A., & Kwon, S. (2008). Guided and fluidic self-assembly of microstructures using railed microfluidic channels. *Nature Materials*, *7*, 581.

Detweiler, C., Vona, M., Yoon, Y., Yun, S.-K., & Rus, D. (2007). Self-assembling mobile linkages. *IEEE Robotics & Automation Magazine*, *14*, 45-55.

Everist, J., Mogharei, K., Suri, H., Ranasignhe, N. K., Will, P., & Shen, W.-M. (2004). A system for in-space assembly. *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 2356-2361). Sendai, Japan.

Fitch, R., & Rus, D. L. (2003). Self-reconfiguring robots in the USA. *Jpn. Robot. Soc. J*, *21* (1), 4–10.

Flow Science, Santa Fe, NM. (2000). *Flow-3D computational fluid dynamics software*. Retrieved May 28, 2009, from http://www.flow3D.com

Fukuda, T., Nakagawa, S., Kawauchi, Y., & Buss, M. (1988). Self organizing robots based on cell structures—CEBOT. *Proceeding of IEEE/RSJ International*

Conference on Intelligent Robots and Systems, (pp. 145–150). Tokyo, Japan.

Gibson, M. A., & Bruck, J. (2000). Efficient exact stochastic simulation of chemical systems with many species and many channels. *Physical Chemistry A*, *104*, 1876–1889.

Gillespie, D. T. (1977). Exact stochastic simulation of coupled chemical reactions. *Physical Chemistry*, *81*, 2340–2361.

Gilpin, K., Kotay, K., Rus, D., & Vasilescu, I. (2008). Miche: modular shape formation by self-disassembly. *International Journal of Robotics Research*, *27*, 345-372.

Glotzer, S. C. (2004). Some assembly required. *Science*, *306*, 419-420.

Goldstein, S. C., Campbell, J. D., & Mowry, T. (2005). Programmable matter. *IEEE Computer*, *28* (6), 99-101.

Gracias, D. H., Tien, J., Breen, T. L., Hsu, C., & Whitesides, G. M. (2000). Forming electrical networks in three dimensions by self-assembly. *Science*, *289*, 1170-1172.

Griffith, S., Goldwater, D., & Jacobson, J. M. (2005). Robotics: self-replication from random parts. *Nature*, *437*, 636.

Grzybowski, B. A., Winkleman, A., Wiles, J. A., Brumer, Y., & Whitesides, G. M. (2003). Electrostatic self-assembly of macroscopic crystals using contact electrification. *Nature Materials*, *2*, 241-245.

Hjelle, D. A., & Lipson, H. (2009). A robotically reconfigurable truss. *Proceedings of ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots.* London, England.

Huang, Y., Duan, X. F., Wei, Q. Q., & Lieber, C. M. (2001). Directed assembly of one-dimensional nanostructures into functional networks. *Science*, *291*, 630-633.

Jacobs, H. O., Tao, A. R., Schwartz, A., Gracias, D. H., & Whitesides, G. M. (2002). Fabrication of a cylindrical display by patterned assembly. *Science*, *296*, 323-325.

Jenekhe, S. A., & Chen, X. L. (1999). Self-assembly of ordered microporous materials from rod-coil block copolymers. *Science*, *283*, 372-375.

Jones, C., & Mataric, M. (2004). Automatic synthesis of communication-based coordinated multi-robot systems. *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 381-387). Sendai, Japan.

Kalontarov, M., Tolley, M. T., Lipson, H., & Erickson, D. (2010). Hydrodynamically driven docking of blocks for 3D fluidic assembly. *Microfluidics and Nanofluidics*, *9*, 551-558 .

Klavins, E. (2007). Programmable self-assembly. *IEEE Control Systems Magazine*, *27* (4), 43-56.

Klavins, E., Ghrist, R., & Lipsky, D. (2006). A grammatical approach to self-organizing robotic systems. *IEEE Transactions on Automatic Control*, *51*, 949-962.

Krishnan, M., Tolley, M. T., Lipson, H., & Erickson, D. (2009). Hydrodynamically tunable affinities for fluidic assembly. *Langmuir*, *25*, 3769-3774.

Krishnan, M., Tolley, M. T., Lipson, H., & Erickson, D. (2008). Increased robustness for fluidic self assembly. *Physics of Fluids*, *20*, 073304-1-073304-4.

Lee, W. H., & Sanderson, A. C. (2002). Dynamic rolling locomotion and control of modular robot. *IEEE Transactions on Robotics and Automation*, *18* (1), 32–41.

Lerman, K., Jones, C., Galstyan, A., & Mataric, A. J. (2006). Analysis of dynamic task allocation in multi-robot systems. *International Journal of Robotics Research*, *25*, 225-241.

Lobo, D., Hjelle, D. A., & Lipson, H. (2009). Reconfiguration algorithms for robotically manipulatable structures. *Proceedings of ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots*. London, England.

Lopes, W. A., & Jaeger, H. M. (2001). Hierarchical self-assembly of metal

nanostructures on diblock copolymer scaffolds. *Nature*, *414*, 735-738.

Maney, K. (2003). *The Maverick and His Machine: Thomas Watson, Sr. and the Making of IBM.* New Jersey: John Wiley & Sons.

Manz, A. G. (1990). Miniaturized total chemical-analysis systems - a novel concept for chemical sensing. *Sensors and Actuators B-Chemical*, 244-248.

McCarthy, W. (2000). Programmable Matter. *Nature*, *407*, 569.

Murata, S., Yoshida, E., Kamimura, A., Kurokawa, H., Tomita, K., & Kokaji, S. (2002). M-TRAN: Self-reconfigurable modular robotic system. *IEEE/ASME Transactions on Mechatronics*, *7* (4), 431–441.

Nagpal, R., Shrobe, H., & Bachrach, J. (2003). Organizing a global coordinate system from local information on an ad hoc sensor network. *Proceeding of International Workshop on Information Processing in Sensor Networks*, (pp. 333-348). Palo Alto, CA.

Napp, N., Burden, S., & Klavins, E. (2006). The statistical dynamics of programmed self-assembly. *Proceedings of the IEEE International Conference of Robotics and Automation.* Orlando, FL.

Objet Geometries Ltd., Billerica, MA. (n.d.). *3D modeling with FullCure® materials*. Retrieved September 15, 2009, from http://www.objet.com/Materials/FullCure_Materials/

Pamecha, A., Chiang, C.-J., Stein, D., & Chirikjian, G. (1996). Design and implementation of metamorphic robots. *Proceedings of the ASME Design Engineering Technical Conference & Computers Engineering and Information Engineering Conference.* Irivine, CA.

Pamecha, A., Ebert-Uphoff, I., & Chirikjian, G. S. (1997). Useful metrics for modular robot motion planning. *IEEE Transactions on Robotics and Automation*, *13* (4), 531–545.

Parviz, B. A., Ryan, D., & Whitesides, G. M. (2003). Using self-assembly for the fabrication of nano-scale electronic and photonic devices. *IEEE Transactions on Advanced Packaging* , *26*, 233-241.

Penrose, L. S., & Penrose, R. (1957). A self-reproducing analogue. *Nature*, *179*, 1183.

Philip, D. a. (1996). Self-assembly in natural and unnatural systems. *Angewandte Chemie*, 1154-1196.

Rothemund, P. W. (2006). Folding DNA to create nanoscale shapes and patterns. *Nature*, *440*, 297-302.

Sagan, C. (1980). *Cosmos*. New York: Ballantine Books.

Smith., R. (2004). *Open dynamics engine*. Retrieved May 28, 2009, from http://www.ode.org

Srinivasan, U., Liepmann, D., & Howe, R. T. (2001). Microstructure to substrate self-assembly using capillary forces. *Journal of Microelectromechanical Systems*, *10*, 17-24.

Stam, J. (1999). Stable fluids. *Proceedings of Special Interest Group on Graphics and Interactive Techniques (SIGGRAPH)*, (pp. 121-128.). Los Angeles, CA.

Stone, H., Stroock, A., & Ajdari, A. (2004). Engineering flows in small devices: microfluidics toward a lab-on-a-chip. *Annual Review of Fluid Mechanics*, *36*, 381-411.

Terada, Y., & Murata, S. (2004). Automatic assembly system for a largescale modular structure - hardware design of module and assembler robot. *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, (pp. 2349-2355). Sendai, Japan.

Thorsen, T., Maerkl, S. J., & Quake, S. R. (2002). Microfluidic large-scale integration. *Science*, *298*, 580-584.

Tolley, M. T., Baisch, A., Krishnan, M., Erickson, D., & Lipson, H. (2008).

Interfacing methods for fluidically-assembled microcomponents. *Proceeding of the IEEE International Conference on Microelectromechanical Systems*, (pp. 1073-1076). Tucson, AZ.

Tolley, M. T., Kalontarov, M., Neubert, J., Erickson, D., & Lipson, H. (2010). Stochastic modular robotic systems: a study of fluidic assembly strategies. *IEEE Transactions on Robotics*, *26*, 518-530.

Tolley, M. T., Krishnan, M., Erickson, D., & Lipson, H. (2008). Dynamically programmable fluidic assembly. *Applied Physics Letters*, *93*, 254105-1-3.

Tolley, M. T., Zykov, V., Erickson, D., & Lipson, H. (2006). Directed fluidic self-assembly of microscale tiles. *Proceedings of μTAS Conference*, (pp. 1552-1554). Tokyo.

Tolley, M., & Lipson, H. (2010). Fluidic Manipulation for Scalable Stochastic 3D Assembly of Modular Robots. *Procedings of the International Conference on Robotics and Automation*, (pp. 2473-2478). Anchorage, AK.

Unger, M. A., Chou, H.-P., Thorsen, T., Scherer, A., & Quake, S. R. (2000). Monolithic microfabricated valves and pumps by multilayer soft lithography. *Science*, *288*, 113-116.

Varshavskaya, P., Kaelbling, L. P., & Rus, D. (2008). Automated design of adaptive controllers for modular robots using reinforcement learning. *The International Journal of Robotics Research*, *27*, 505-526.

Varsos, K., & Luntz, J. (2006). Superposition methods for distributed manipulation using quadratic potential force fields. *IEEE Transactions on Robotics*, *22*, 1202-1215.

Vilkner, T., Janasek, D., & Manz, A. (2004). Micro Total Analysis Systems. Recent Developments. *Analytical Chemistry*, *76*, 3373-3385.

Vlasov, Y. A., Bo, X.-Z., Sturm, J. C., & Norris, D. J. (2001). On-chip natural

assembly of silicon photonic bandgap crystals. *Nature*, *414*, 289-293.

Werfel, J., & Nagpal, R. (2008). Three-dimensional construction with mobile robots and modular blocks. *International Journal of Robotics Research*, *27*, 463–479.

Werfel, J., Bar-Yam, Y., Rus, D., & Nagpal, R. (2006). Distributed construction by mobile robots with enhanced building blocks. *Proceedings of International Conference on Robotics and Automation*, (pp. 2787-2794). Orlando, FL.

Whang, D., Jin, S., Wu, Y., & Lieber, C. M. (2003). Large-scale hierarchical organization of nanowire arrays for integrated nanosystems. *Nano Letters*, *3*, 1255-1259.

White, P. J., Kopanski, K., & Lipson, H. (2004). Stochastic self-reconfigurable cellular robotics. *Proceeding of the IEEE International Conference on Robotics and Automation*, *3*, pp. 2888-2893. New Orleans.

White, P., Zykov, V., Bongard, J., & Lipson, H. (2005). Three dimensional stochastic reconfiguration of modular robots. *Proceedings of Robotics Science and Systems* , (pp. 161-168). Cambridge.

Whitesides, G. M., & Grzybowski, B. (2002). Self-assembly at all scales. *Science*, *295*, 2418-2421.

Winfree, E., Liu, F., Wenzler, L. A., & Seeman, N. C. (1998). Design and self assembly of two-dimensional DNA crystals. *Nature*, *394*, 539–544.

Yang, P. D., Deng, T., Zhao, D. Y., Feng, P. Y., Pine, D., Chmelka, B. F., et al. (1998). Hierarchically ordered oxides. *Science*, *282*, 2244-2246.

Yim, M., Shen, W.-M., Salemi, B., Rus, D., Moll, M., Lipson, H., et al. (2007). Modular self-reconfigurable robotic systems. *IEEE Robotics and Automation Magazine*, *14* (1), 43–52.

Yim, M., Zhang, Y., & Duff, D. G. (2002). Modular robots. *IEEE Spectrum*, *39* (2), 30–34.

Zykov, V., & Lipson, H. (2007). Experiment design for stochastic three-dimensional reconfiguration of modular robots. *Proceedings of the IEEE International Conference on Intelligent Robots and Systems, Self-Reconfigurable Robotics Workshop.* San Diego, CA.

Zykov, V., Mytilinaios, E., Desnoyer, M., & Lipson, H. (2007). Evolved and designed self-reproducing modular robotics. *IEEE Transactions on Robotics*, *23* (2), 308–319.